Dense Surface Models of the Human Face

Tim J. Hutton
Biomedical Informatics Unit, Eastman Dental Institute,
University College London

A thesis submitted for the degree of Doctor of Philosophy.

September 17, 2004

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

This thesis describes and evaluates *Dense Surface Models* (DSMs), a new technique for building point distribution models of surfaces, from raw input data. DSMs can be used on data from a wide range of surface acquisition systems without preprocessing since they do not require that the surfaces be closed or even locally manifold, and can cope well with holes and spikes in the surfaces. This is an advantage over comparable techniques, which impose such constraints on the input.

The core of the DSM algorithm is as follows. A dense correspondence is made between the surfaces using thin-plate spline warping guided by means of a small set of hand-placed landmarks. The area of interest is automatically defined by a threshold on a measure of the closeness of the correspondence at each point. A point distribution model is then built using the vertices from the trimmed and densely-corresponded surfaces.

The key benefit of using models of the whole surface is illustrated by the large improvement in classification on face shape that is obtained when using DSMs as compared to landmark-based geometric morphometrics. This is demonstrated by testing classification by gender and also by congenital anomaly where facial growth and form is abnormal. The latter is currently the primary application of DSMs.

The use of DSMs for automatically fitting to new scans is evaluated for robustness and accuracy. Methods for analysing continuous and discrete parameters such as age and gender are presented and evaluated. The incorporation of grey-level information with the shape information is also possible, and is explored.

Contents

Publications

Work from this thesis has appeared in the following co-authored publications:

Theory:

- Hutton T.J., Buxton B.F., Hammond P. (2001) *Dense surface point distribution models of the face.* Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, Kauai, Hawaii: 153-160.

- Hutton T.J., Buxton B.F., Hammond P. (2002) *Estimating Average Growth Trajectories in Shape-Space using Kernel Smoothing.* In: Jon Sporring et al. (eds.), Proceedings of the International Workshop on Growth and Motion in 3D Medical Images, European Conference on Computer Vision, 1st June 2002, Copenhagen, Denmark. pp. 1-7.

- Hutton T.J., Buxton B.F., Hammond P., Potts H.W.W. (2003) *Estimating Average Growth Trajectories in Shape-Space using Kernel Smoothing.* IEEE Transactions on Medical Imaging 22(6): 747-753.

- Hutton T.J., Buxton B.F., Hammond P. (2003) *Automated Registration of 3D Faces using Dense Surface Models.* In: Harvey R. and Bangham J.A. (Eds.), Proceedings of the British Machine Vision Conference, Norwich. pp. 439-448.

Applications:

- Hammond P., Hutton T.J., Patton M.A., Allanson J.A. (2001) *Use of 3D Photogrammetry in the Craniofacial Assessment of Noonan Syndrome.* XXII David W. Smith Workshop on Malformations and Morphogenesis, UCLA Conference Centre, Lake Arrowhead, CA, USA, Sept 7-12, 2001.

- Hammond P., Hutton T.J., Patton M.A., and Allanson J. (2001) *Delineation and Visualisation of Congenital Abnormality using 3D Facial Images.* In: Bellazzi R., Zupan B., and Liu X. (Eds.), Proceedings of the Workshop Intelligent Data

Analysis in Medicine and Pharmacology, IDAMAP2001 at MedInfo2001, London, UK. Pages 26-29.

- Hammond P., Hutton T.J., Allanson J.A., Shaw A., Patton M.A. (2002) *3D Digital Stereophotogrammetric Analysis of Noonan Syndrome.* British Human Genetics Conference 2002, York. J Med Genet 39: Supplement 1, S35.

- Hammond P., Hutton T.J., Shaw A., Temple I.K., Hennekam R.C.M., Winter R.M., Patton M.A., Allanson J.E. (2003) *Modelling Facial Form and Growth in Noonan Syndrome in 3D.* D.W. Smith Meeting on Malformation and Morphogenesis, UBC, Vancouver, Canada, August 2003.

- Hammond P., Hutton T.J., Allanson J.A., Smith A.C.M. (2003) *The 3D face of Smith-Magenis syndrome (SMS): a study using dense surface models.* European Human Genetics Conference 2003, Birmingham. Eur J Hum Gen 11 (S1), 102.

- Hammond P., Hindocha, N., Hutton T.J., Beales, P.L. (2003) *3D dense surface modelling defines a characterisic facial phenotype in Bardet-Biedl syndrome.* American Society for Human Genetics Conference, Los Angeles. Am J Hum Gen 73 (5) S1, 284.

- Hammond P., Hutton T.J., Maheswaran S., Modgil S. (2004) *Computational models of oral and craniofacial development, growth and repair.* Adv Dent Res 17, 61-64.

- Hammond P., Hutton T.J., Allanson J.E., Campbell L.E., Hennekam R.C.M., Holden S., Patton M.A., Shaw A., Temple I.K., Trotter M., Murphy K.C., Winter R.M. (2004) *3D Analysis of Facial Morphology.* American Journal of Medical Genetics Part A 126(4): 339-348.

Acknowledgments

This thesis wouldn't have been produced without the support of many people, including:

- My supervisors: Peter Hammond, without whom none of this would have happened, and Bernard Buxton, who put me on the path towards image processing in the first place.
- My Mum and Dad, of course. It was my Dad's Radio Shack TRS-80 Model II that first opened my eyes to the fun you could have programming computers to do what *you* wanted them to do.

```
I feel a draft.
You are in Room # 19.
Tunnels lead to 11,18,20.
Shoot or move?
```

- Deepa, for being Fig. 1.1 and so much more.
- My friends, especially the Tongs, for keeping me sane while I was writing up. Rick sent me this, for example:

> Here's a section for you to include, to save you time:
>
> > "Faces are like snowflakes: every one is different. And like snowflakes, the face's structure is based upon a few simple rules. However, unlike snowflakes, faces are not made of ice, and therefore they don't melt, although what I am trying to do is make a computer what can spot faces that HAVE 'melted' - so to speak - by spotting variations from the normal rules. To achieve this, I have spent the past three years scanning the faces of snowmen. I am beginning to suspect this was a mistake."
>
> Arbeit macht frei,
> Rick

Chapter 1

Introduction

This thesis is about a technique for modelling complex biological surfaces such as the human face: *dense surface models* (DSMs). The problem of modelling such variable objects consists of a *correspondence problem* (how to map each point on one face to another) and a *modelling problem* (how to learn a high-dimensional distribution from a small number of training examples).

The DSM algorithm can build surface models from raw data where there can be holes in the surface and other errors, without preprocessing. The approach requires hand-placed landmarks to be supplied to obtain the correct correspondence. While techniques exist that can build surface models without using landmarks (Davies et al., 2002b), these currently require any holes to be filled and any problems in the surface to be corrected. Techniques exist for doing this but manual intervention may be required.

Figure 1.1 shows a typical example of such raw data, captured by a stereo photogrammetry system.

Figure 1.1: An example textured surface scan (left), also shown in wireframe (right).

The surface in Fig. 1.1 is stored and processed as a *polygonal mesh*. While this is

not the only possible way to represent surfaces it is the most commonly used method. Figure 1.2 highlights the holes that can be present in these meshes. These scans also illustrate another potential problem, which is that the extent of the surface area captured can vary dramatically, with clothing, hair, ears, etc. being either present or absent. Any approach to modelling the human face that uses data from surface scanners must be able to deal with this issue in some way.

It should be noted that these issues are not related to any particular design of acquisition system; surface scanners will always be limited by occlusion, causing holes in the nose or other parts that cannot be seen from different angles.

Figure 1.2: Two typical scans from stereo-photogrammetry acquisition systems, shown as a surface *(left)* and as a mesh *(right)*. The scans show variation in the extent covered as well as occasional holes.

If holes occur on smoothly curving parts of the surface, such as the cheek, then interpolation algorithms will be able to fill them reliably (eg. Carr et al., 2001). However, holes in noisy parts of the surface, such as eyebrows, present greater difficulties.

Some other problems that can occur are illustrated in Fig. 1.3. A polygonal surface is *locally manifold* if every non-boundary edge is shared by exactly two polygons (see eg. Lindstrom and Turk, 1998). If an algorithm requires that the input surfaces have this property then some pre-processing step will often be necessary. In general this is

a very difficult problem.

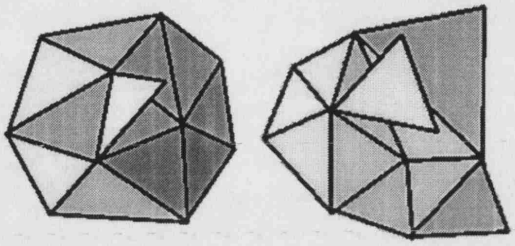


Figure 1.3: Two examples of surfaces that are not locally manifold (see text). On the left a triangle is connected to the surface by only one edge, while on the right by only one vertex. The other (non-boundary) polygons are locally manifold since every edge is shared by exactly two polygons.

The degree to which such problems are seen in surfaces depends of course on how the data has been acquired. Iso-surfaces from volume images (MRI or CT, for example), will not have holes but may have unconnected regions, which also present a problem to some modelling algorithms. Laser-scanners and stereo-photogrammetry systems typically have difficulty imaging hairy surfaces such as the top of the head and the eyebrows. Wet surfaces such as the eyeball and sometimes the lips can also present a problem to the scanner, typically leaving holes or other errors on the surface. Additionally, holes will be caused by any structures that cause occlusion, such as the nostrils or the folds of tissue in the ear.

Our approach copes with all of the above issues since it does not traverse the surface (working from polygon to neighbouring polygon) at any stage.

The traditional approach to analysing the shape of biological objects is geometric morphometrics (see eg. Dryden and Mardia, 1998; O'Higgins and Jones, 1998), where the positions of landmarks placed on a population of examples are compared. The key technical observation behind the development of the DSM algorithm is that even if many landmarks are placed on a surface such as a face, there will still be shape information in the parts of the surface between the landmarks, and this information is likely to be relevant to the problem at hand.

1.1 Motivation

The clinical motivation for wanting to analyse the shape of the human face is that the face is an indicator of several medical conditions. In particular, *syndromes* that have an associated facial dysmorphology, such as Noonan Syndrome (Fig. 1.4) and Williams Syndrome, are often diagnosed in children from their facial appearance. The

distinctive features with these syndromes are not always obvious to a layperson but an experienced clinical geneticist can spot them simply by looking at the face. While there are usually other ways to detect these syndromes, such as listening for a heart condition, identifying dysmorphology in other parts of the body, or genetic testing, the face is very accessible, and taking photographs or 3D scans of the face is a rapid and non-invasive procedure.

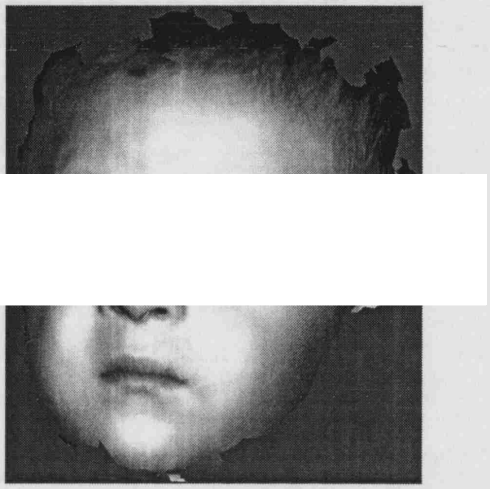


Figure 1.4: A boy with Noonan Syndrome. While the differences associated with the syndrome are not obvious, they include: the outer corners of the eyes are lower than the inner corners, the eyes are further apart than usual, the bridge of the nose is depressed, the mouth is small, the lower face is underdeveloped and triangular.

The clinical application of face shape analysis could take two forms. Firstly, a population of syndromic and non-syndromic faces could be analysed to extract the differentiating features, and these could be presented visually, perhaps as animations from one extreme to another. By extracting the important features and getting rid of the irrelevant variation such as facial expression, gender variation and individual face differences, the junior clinician can be trained in the art of recognising the condition without having to examine many face images. Clinical geneticists have already commented on how useful the synthesised images and animations have been when they were presented at meetings.

A second clinical application of face shape analysis is to support diagnosis directly through the automatic classification of a scan of a subject. Computer-assisted screening could best be used where experienced clinicians are not available, in developing countries, for example.

Technologically, there have been several developments that make analysing 3D face shape now possible. Computer vision research on stereo photogrammetry has produced algorithms that can take images from commonly available digital cameras and compute

the three-dimensional surface of the object being imaged. Complete systems that combine the hardware needed to position and calibrate the cameras and the software needed to process the images are now commercially available. Some of these systems are capable of capturing face shape instantaneously, an important consideration when the subjects are young children or have the behavioural difficulties that are associated with certain syndromes. Additionally, the processing power, storage requirements and hardware graphics support necessary for visualising and processing surface scans are now available in standard desktop PCs.

1.2 Aims

In this thesis we aim to show the following:

> *It is possible to build dense surface models from surface scans of varying shape that include small holes and other errors, when supplied with a small set of hand-placed landmarks.*

To clarify the descriptive terms used:

- By 'possible' we mean that using an implementation of the algorithm in some standard computer language we can run the program in a reasonable amount of time (hours at most) on standard computer equipment such as desktop PCs.
- The surface model is 'dense' in the sense that the surface can be represented (if desired) with an average edge length smaller than the size of the smallest features. Such a surface mesh is able to represent the curving surface of a face as accurately as required.
- The input surfaces can vary in shape but there are limits to the amount of variation that can be modelled. The surfaces must vary continuously (although this requirement could be relaxed).
- The surfaces that are used as input may include small holes, where by 'small' we mean only a few times the average polygon size. Although this is not a precise definition, it is shown that the model can work in the presence of small holes, and it is suggested that the performance of the model will degrade gracefully in the presence of increasing numbers of holes, or holes of increasing size.
- The algorithm requires a 'small' set of landmarks, where by small we mean far fewer (tens) than the number of vertices in the final model (typically many thousands), and only as many as there are reproducibly identifiable points on the surface, which for the face is between around 9 and 20 (Gwilliam, 2004).

Subsidiary aims are tested in Chapters 5-7 as follows:

Chapter 5: *Dense surface models can be used automatically to register unseen face scans to within an accuracy comparable to that of manual landmark placement.*

Chapter 6: *Dense surface models can be used to classify unseen scans in categories learnt from the training set, and give superior performance compared to an approach based only on the landmarks.*

Chapter 7: *Surface models that combine shape and texture information can be built, even from scans where the texturing method means that the object may not be contiguous in the texture image.* This issue is somewhat subtle and since it is not connected to the rest of the thesis it is explained in the chapter rather than here.

1.3 Contributions

The core contribution of this thesis is the dense surface model algorithm, for building useful surface models directly from a landmarked set of scans. While there are other algorithms for building surface models, DSMs can take the type of surface typically captured on surface-acquisition systems and directly (ie. without pre-processing) make a dense correspondence between the surfaces (using a small set of hand-placed landmarks), and trim the surfaces to include only those parts that are well-corresponded. This makes it immediately useful for analysing databases of 3D face scans, as are being collected by many organisations.

Additionally, we present here methods for fitting the dense surface template to unseen scans automatically, using a hybrid of active shape model (Cootes et al., 1995) and iterative closest point (Besl and McKay, 1992) fitting. This method means that new scans can be analysed in a fully automatic fashion, without needing to be manually annotated with landmarks before analysis.

The third contribution is to show how textured surface models can be built, by computing shape-free versions of the texture images of the scans (following Cootes et al., 1998). The innovation here is that for a surface textured with images taken from multiple viewpoints, the object does not appear in a contiguous area in any one texture image. Thus, normal warping methods for producing the shape-free images cannot be used.

The dense surface model algorithm is fully implemented, and the software is in constant use within our department in conjunction with a dozen international clinical

collaborators. The main application at the time of writing is the analysis of the facial features associated with genetic syndromes, supporting work by clinical geneticists. One copy has been sold, to a defence research contractor. Recently the software has been made available to a research hospital in Hangzhou, China, as part of an NIH-funded international project screening undiagnosed children with moderate to severe learning disability.

1.4 Structure of the thesis

Chapter 2 gives some technical background to the area of working with surfaces and shape analysis and discusses the prior art.

Chapter 3 gives details of how DSMs are built, demonstrates how the algorithm copes with holes and spikes and other artefacts, and shows some initial results.

Chapter 4 looks at how well the model generalises, by testing its sufficiency and specificity in modelling new scans.

Chapter 5 demonstrates the use of DSMs to register automatically unseen surface scans, and compares the accuracy of landmark placement obtained by this method with the accuracy of human landmark placement.

Chapter 6 evaluates the use of DSMs for classifying new scans into learnt categories. In this chapter we give the main clinical application of DSMs, which is to screen for the presence of genetic syndromes with associated dysmorphic facial features, the diagnosis of which is often not straightforward.

Chapter 7 shows how DSMs can be augmented with the colour information acquired at each point on the surface scan, making photorealistic models that again have a clinical application in assisting the diagnosis of facial dysmorphic syndromes.

Chapter 8 brings the main points of the thesis together as conclusions and highlights possible extensions and improvements of the work that could be attempted in the future.

Chapter 2

Background

In this chapter we look at how this thesis fits in with the body of existing work, after introducing some necessary concepts.

2.1 Surface representations

The surface of a three-dimensional object can be represented in many ways. For a simple sphere, the location of the centre and a radius are enough to describe its curved shape, while a cube can be described using six planes. For more complicated objects such as are encountered in medical imaging there are several possible representations, including (Fig 2.1):

a) a *mesh* of many small facets covering the object,

b) a *volume* of small regions each marked as either inside or outside the object, or

c) a combination of different distortions of a sphere, described as *spherical harmonics.*

The raw data on which we have been working has typically been in the first of these three forms, consisting of a series of vertices and a set of triangles joining them. CT and MRI scanners, by contrast, output volume data in the form of intensities at every point on a regular 3D grid. In general, the three representations above are interchangeable since for many surfaces there exist operations to transform between one representation and another. One exception to this is a surface that contains one or more holes, causing it to be *open* - it is not possible to represent this type of surface in the form of a volume or with spherical harmonics without filling in the hole. Similarly, surfaces with disconnected parts cannot be represented using a single spherical harmonics representation. For the remaining part of this thesis we shall concentrate exclusively on the polygonal mesh representation unless specifically indicated.

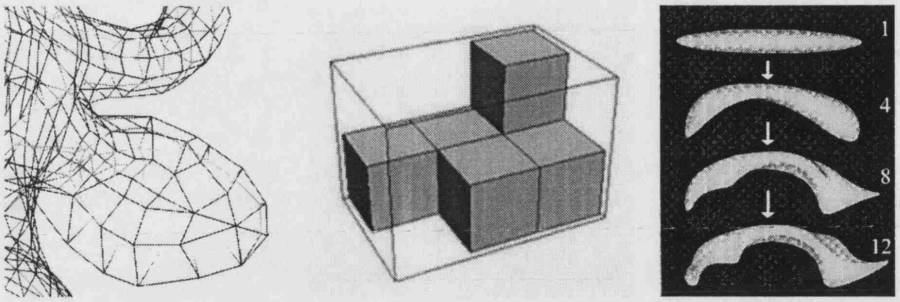


Figure 2.1: Three representations of surfaces; a polygonal mesh *(a)*, a volume image *(b)*, and a set of spherical harmonics *(c)* (adapted from Gerig et al., 2001).

2.2 Geometric operations on surfaces

As three-dimensional objects, surfaces can be manipulated in the normal ways. *Linear transforms* include translation, rotation and scaling. These are all operations that take place on the vertices of the mesh alone, since the connectivity of the cells remains unchanged. Any set of such linear transforms can be encapsulated in a 4×4 transform matrix, by making use of homogeneous coordinates. Reflection and shearing are also possible within such a framework but since they are not operations that typically occur on concrete objects in the natural world we will leave them aside for now.

One of the many ways to specify a linear transformation is to give two sets of points and use the transform that maps one to the other. In general there will not exist a linear transform that exactly maps the points onto one another and so an error term must be minimized to find the best solution. We will make use of this *landmark transform* at many stages, the algorithm we use is given in Horn (1987) and implemented in the VTK class vtkLandmarkTransform.

Another operation that we use frequently is the ability to query a surface for its closest point to a given location. Together with the landmark transform this forms the basis of another algorithm - the Iterated Closest Point algorithm (Besl and McKay, 1992) that can be used to rigidly align two surfaces. The operation of ICP is as follows:

1. for all the vertices in one surface, find the closest point on the second surface, and then
2. compute the landmark transform that best maps the vertices to their closest points,
3. apply the landmark transform to the first surface, and
4. repeat from 1 until no there is no further movement.

Figure 2.2 shows the ICP algorithm in action on two faces.

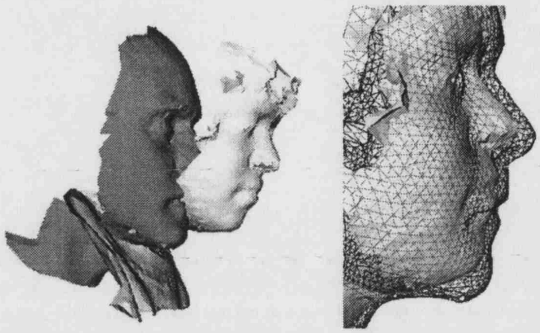


Figure 2.2: The Iterated Closest Point algorithm in action, registering two faces. The initial configuration *(left)* and the final result *(right)*.

ICP is just one algorithm of many in the field of *medical image registration*, where the aim is to find the correspondences between, for example, two images of the same patient, or between an image of a patient and an atlas of labelled medical parts. One key issue is whether the registration is rigid or non-rigid - ICP performs rigid registration since it uses a linear transform. *Non-linear transforms*, on the other hand, can change not only the position and orientation of objects but also their shape. This is by definition, since we define shape to be the properties of the object that are dependent on the position of its parts but invariant to linear transforms - two objects have the same shape if there exists a linear transform that exactly maps one onto the other.

One particular non-linear transform that is useful is the Thin-Plate Spline (Bookstein, 1997b). As with the landmark transform, a TPS transform is specified by two sets of corresponding points. With TPS, one of the sets of landmarks is exactly mapped onto the other and a deforming transform is interpolated between them. The interpolation is chosen such that it minimizes a bending energy, ensuring that the deformation is smooth and no discontinuities appear. For n landmark points $\mathbf{x}_i$ in d dimensions the warping function $\mathbf{f}(\mathbf{x})$ is given by:

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{u}_d(\mathbf{x}) \qquad [d \times 1] \tag{2.1}$$

where $\mathbf{W}$ is a $d \times (n + d + 1)$ matrix of weights and $\mathbf{u}_d(\mathbf{x})$ is given by:

$$\mathbf{u}_d(\mathbf{x}) = (U(|\mathbf{x} - \mathbf{x}_1|), \ldots, U(|\mathbf{x} - \mathbf{x}_n|), 1, \mathbf{x}^{\mathrm{T}})^{\mathrm{T}} \qquad [(n + d + 1) \times 1] \tag{2.2}$$

where U is the basis function. In 3D we simply use $U(r) = r$ while in 2D we use $U(r) = r^2 log(r)$. To find the necessary weights $\mathbf{W}$, construct the matrices:

$$\mathbf{Q} = \begin{pmatrix} 1 & \mathbf{x}_1^{\mathrm{T}} \\ 1 & \mathbf{x}_2^{\mathrm{T}} \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^{\mathrm{T}} \end{pmatrix} \qquad [n \times (d+1)] \tag{2.3}$$

$$\mathbf{L} = \begin{pmatrix} \mathbf{K} & \mathbf{Q} \\ \mathbf{Q}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \qquad [(n+d+1) \times (n+d+1)] \tag{2.4}$$

where $\mathbf{K}_{ij} = U(|\mathbf{x}_i - \mathbf{x}_j|)$ and $\mathbf{0}$ is an array of zeroes. We also construct a matrix of the landmark points in the target image:

$$\mathbf{X}' = \begin{pmatrix} \mathbf{x}_1' \\ \vdots \\ \mathbf{x}_n' \\ \mathbf{0} \end{pmatrix} \qquad [(n+d+1) \times d] \tag{2.5}$$

Then $\mathbf{W}$ is given by the solution to:

$$\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} = \mathbf{X}' \tag{2.6}$$

Figure 2.3 shows the TPS warping of a face.

Figure 2.3: The Thin-Plate Spline (TPS) algorithm demonstrated on a face. Nine landmarks are placed on the face *(left)* and one of them is displaced downwards and slightly forwards *(right)*.

The advantage of using a non-linear transform for registration is that a better correspondence can be achieved when the objects have different shapes - notice how the faces in Fig. 2.2 still differ considerably even though they are rigidly registered. If we instead register the two using a set of landmarks and a TPS transform, we can achieve a much closer correspondence. This is demonstrated in Fig. 2.4 where 9 landmarks are used to warp one face onto another. It is the good performance of this technique

for bringing two faces into close alignment that has motivated the development of our dense surface models algorithm.

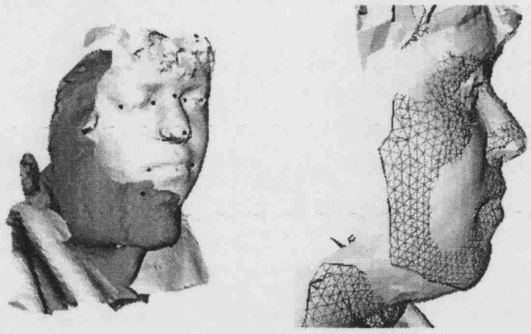


Figure 2.4: TPS-alignment of two faces using nine corresponding landmarks. The initial surfaces *(left)* and the final result *(right)*.

Such a method of registering surfaces can be described as *manual* since it relies on the placing of recognised landmarks by an operator. Automatic methods of registration by contrast require no landmarks but allow the possibility of failing to find the correct correspondence. Manual registration avoids this possibility but of course requires time to landmark each example.

There is a great deal of existing work on automatic image registration methods. Maintz and Viergever (1998) review general techniques including those for registering volume images while Audette et al. (2000) review techniques specific to the registration of 3D surfaces. The many automatic registration methods can be categorised according to:

- the type of transforms permitted:
  - globally linear (Besl and McKay, 1992),
  - globally polynomial (Subsol et al., 1994),
  - locally affine (Feldmar and Ayache, 1994),
  - locally polynomial, eg. Bookstein (1997b),
  - physical constraints based on:
    - FEM models, eg. Pentland and Horowitz (1991),
    - fluid models (Lester et al., 1999);
- the criteria used to make matches between the surfaces:
  - nearest point (Besl and McKay, 1992),
  - point-features (eg. peaks) (Goldgof et al., 1988; Thirion, 1994),

– ridge-line correspondence (Guéziec and Ayache, 1994).

None of these methods uses a model of the class of object being registered, they are general algorithms suited for use on any type of data. In the next sections we introduce the concept of a model. Having a model of the objects of interest (eg. the face) is enormously powerful, allowing the types of transforms permitted to be greatly extended and also removing the necessity to identify features on the target surfaces.

2.3 Statistical models

The classical example of a statistical model is that of the outcome of throwing an ideal six-sided die. The result of a throw is always a value between 1 and 6 and the six cases are equally likely: $p_i = \{\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\}$. This form of model describes the possible outcomes of actually throwing the die by giving the probability of each outcome. For an actual die we could approximate this model by taking lots of samples and accumulating their relative occurence. If the die gave some numbers more than others because it was weighted then this would (eventually) show up in the computed model. We might find, for example, $p_i = \{0.2, 0.2, 0.2, 0.2, 0.1, 0.1\}$, ie. 5 and 6 crop up half as often as the other numbers. Having such a model would be very handy if money were involved.

For continuous variables such as the height of a person we can plot the distribution of different heights on a graph, showing the probabilities across the range. In the case of people's heights there is a very familiar statistical model that can describe the situation - the normal distribution. A normal distribution with a mean of $5'8\frac{1}{2}''$ and a standard deviation of $3''$ tells us that we expect 30% of the population to be taller than $5'10''$ and 70% to be shorter[1]. Notice that unlike the discrete case of the die we cannot give a probability for each outcome (what are the odds of someone being *exactly* $5'10''$?), but we can make statements about ranges of heights and about the relative occurence of different heights.

Such statistical models correspond well to the concept of a 'mental model'. If we saw someone taller than $6'6''$ we might be surprised and rightly so because less than 0.1% of the population is taller than that height. Satisfyingly, this intuitive concept extends neatly to more complicated structures such as the human face - if we see a face that is outside the 'normal' range then it looks odd; our brains have a very sophisticated mental model of human faces. Of course a face cannot be described with a single number as can the height of a person, instead we have to use multiple parameters, and the statistical model we build is a multivariate one.

In multivariate statistics, each example of a class of objects is represented by a vector of parameters or measurements and the aim is often to find correlations between

[1]For men in the US. Data from `http://www.shortsupport.org/Research/analyzer.html`

combinations of the variables. As with single variable models like the dice and heights examples, with a multivariate model we create a representation of how the probability of occurence changes over the space of the parameters. As we venture into higher numbers of dimensions the familiar normal distribution transfers but is transformed into a (hyper-)ellipsoid, with examples that have a higher probability of occurence being found in the centre of the ellipsoid and less likely examples being found further out. As with the weighted die we can infer a multivariate model by taking many samples of the possible outcomes.

2.4 Shape models

One particular multivariate model that has proven to be especially useful for studying biological objects is the *point distribution model* (Cootes et al., 1992). Here, an imaged object is annotated by placing landmarks around its border or on other recognisable points. Each landmark has a horizontal and a vertical coordinate, the set of landmarks makes up a 'template'. By concatenating the x and y coordinates for each landmark into a vector we can represent every template with a single location $\mathbf{x}$ in a $2n$-dimensional space:

$$\mathbf{x} = [x_1, y_1, x_2, y_2 \ldots x_n, y_n]^{\mathrm{T}} \tag{2.7}$$

If the object is three-dimensional, then the space is $3n$-dimensional, etc.

If we are interested in modelling the different shapes of the templates then it is normal to remove the spurious effects of the template location and orientation before transforming the examples into the space. This can be done by using the Procrustes algorithm (Gower, 1975; Goodall, 1991) which we introduce in the next chapter.

Some areas of the space spanned by these vectors (the *shape-space*) will be more densely populated than others, some will be completely empty because the vectors there do not correspond to real instances of the class of objects. By taking lots of examples of the input class we can infer a model of the distribution, the point distribution model. One way to represent the varying probabilities is to find the (multivariate) normal distribution that best models the data and in practice this works very well, learning from the training set which templates are legal examples and which are not.

One function of such a model is to allow new examples of the object to be synthesised. This can be done simply by choosing a location in the space that has a probability above a certain threshold, say within three standard deviations of the mean.

2.5 Surface models

The core algorithm of this thesis is a combination of two of the concepts we have introduced: surface correspondence and shape models. In the following chapter we show how we can build a *dense surface model* from a set of surfaces - a shape model that uses as its template a dense set of landmarks across the surface, extracted from the registered inputs.

Several groups have undertaken relevant work concerning dense correspondences between two or more 3D surfaces. These are briefly reviewed below.

Brett et al. (1997); Brett and Taylor (1998, 2000) is a series of work on making dense correspondences between surfaces, using a 'brushfire' algorithm. The approach suggested is to first establish a rigid correspondence between pairs of shapes, making use of a highly decimated version of each, then to fill-in the dense vertices using a brushfire algorithm across each surface. The method requires the surfaces to be sufficiently similar in shape that a rigid-body match will converge to give the correct correspondence. For classes of object that exhibit large shape changes it is not clear that the method will continue to give good results. Also, they demonstrate their method on a set of closed surfaces but do not explain how the method could make use of open surfaces such as the human face where the boundary of the area of interest is poorly defined and the extent of the input data varies from example to example.

Blanz and Vetter (1999) make dense correspondences between cylindrical scans, taking advantage of the fact that the radial coordinates from Cyberware scans can be expressed as a height map. This renders their technique less generally applicable than might otherwise be the case, although their results are very convincing. They use optical flow techniques automatically to establish correspondences between texture images and between height-maps as intensity images. Paterson and Fitzgibbon (2003) reimplement this work but opt to use a radial basis function (RBF) warp with manual landmark placement to bring the surfaces into correspondence, citing the poor reliability of optical flow as the reason not to use it.

Lee et al. (1999) present work that is similar to the approach of Brett and Taylor (1998). They first establish a correspondence between greatly simplified meshes (15 vertices or so) and then extend the correspondence to higher-resolution meshes. User-intervention is required to maximise the film-quality of the final correspondence since their application was animated morphing between surfaces for the entertainment industry.

Lorenz and Krahnstöver (2000) show an improved method for building dense surface models that is similar to ours but which (as in Brett and Taylor, 1998) is only demonstrated on closed surfaces. All the input meshes are warped onto the hand-placed landmarks of a single example, then a coating procedure is used to resample

each surface to solve the correspondence problem. A mesh regularization step is necessary to ensure that folds in the surface introduced by the coating procedure do not appear in the final model. They demonstrate their method on a training set of 31 lumbar vertebrae, using 15 landmarks to produce a point distribution model with approximately 600 vertices. As in our work, this approach sacrifices full automation at the model-building stage in exchange for more robustness to large deformation, since it allows the user to annotate whatever shape changes are necessary to bring surfaces into alignment.

Andresen et al. (2000) use surface-bounded diffusion automatically to find correspondences between mandibles extracted from CT scans of patients with Apert syndrome. This algorithm requires a volume-image representation, making it suitable for surfaces in CT scans but not suitable for open surfaces without pre-processing.

Wang et al. (2000) make a dense correspondence between surfaces by using surface features based on curvature. Using shape to establish the correspondence is obviously problematic, as pointed out by Tagare (1999) (mentioned in Brett and Taylor, 2000). However, where the shape change is small relative to the shape features being used this technique could be used.

Davies et al. (2002a,b) present a method for building 3D surface models using description length minimisation. The dense correspondence is manipulated by using a set of distortions which move the vertices around, and the correspondence which gives the most compact final model is chosen. The vertices are not allowed to collapse to a single point as this would represent a trivially compact solution. While this method is attractive for several reasons, for example it is fully automatic and gives a reasonable criterion for the optimal model, applying this method to face scans presents several problems. Firstly, the face scans would require preprocessing to fill their holes and strip out any loose or partially-connected triangles, since the surfaces would have to be locally manifold at all points (except the boundary) to be used with a minimum description length (MDL) approach. Secondly, some method for restricting the extent of the surface would be required. It should be possible to use a hand-crafted base mesh as a master example but this has not yet been shown to work for 3D surfaces. It seems likely that all of these problems can be solved, and it would be a useful contribution to do so. Methods for more efficiently finding the best correspondence have been suggested (Ericsson and Åstrøm, 2003).

Rueckert et al. (2003) show how statistical deformation models (SDMs) can be automatically constructed for volume images of the brain by computing free-form deformations (FFDs) that maximise the normalised mutual information. FFDs are a method for nonrigid warping with local control, through the use of cubic B-splines. By contrast, TPS does a global nonrigid warping controlled by a set of landmarks.

Chapter 3

Dense Surface Models Overview

In this chapter we give details of how DSMs are built. The properties of DSMs are evaluated in later chapters.

3.1 Input data

The data with which we are working are surface scans. Typically these are triangular meshes, and may or may not have a texture image associated with them. The majority of the surfaces used in this thesis were captured on a DSP400 machine made by 3dMD (`www.3dMD.com`). The DSP400 system uses stereo photogrammetry to build a 3D mesh from a set of two or more 2D images of an object, taken from different (known) angles. More recently, scanners from MedEIM (`www.medeim.com`), the marketing branch of SurfIm (`www.surfim.com`), have been used.

The exact workings of the acquisition system are not of concern since our algorithm works only with the surfaces that are output. Alternative sources of suitable data include:

- laser scanning systems:
  - the Minolta VIVID range,
  - the Polhemus hand-held scanner,
  - UCL's own laser-imaging system;
- other range-finding systems:
  - Hamamatsu's body-scanner;
- iso-surfaces from volume images: (the application of a contouring algorithm such as Marching Cubes (Lorensen and Cline, 1987) on a volume image gives a polygonal surface)

- CT or MRI images,
- 3D confocal microscopy images,
- ultrasound/sonar/radar data.

While the DSM algorithm was developed for human faces it can be applied to any type of surface data where there is a natural correspondence. Typically this means biological subjects, since the underlying mechanism behind the production of the surfaces is the same in all cases, the differences being due to developmental or genetic variation. By contrast, machine parts or artificial structures like buildings have discontinuous shape variation and while techniques exist for analysing discontinuous distributions this is outside of the scope of this thesis. In the remaining part of this chapter and in subsequent chapters we shall refer almost exclusively to human faces as the objects being studied.

3.1.1 Acquisition protocol

The MedEIM FCS cameras are positioned in front of and slightly below the subject (Fig. 3.1), in order that all of the structures of the face are visible to the cameras, including underneath the nose and chin.

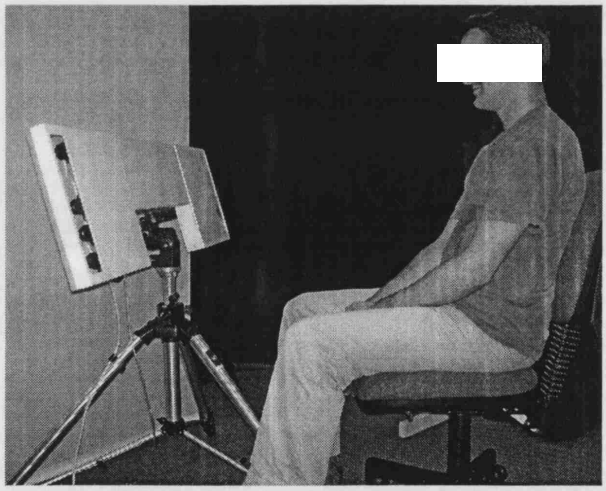


Figure 3.1: A subject being scanned on the MedEIM FCS (face capture system). Six cameras in total capture images at the same instant.

For the main study, the subject is instructed to look straight forwards, and if they are obviously smiling or making some other facial expression they are asked to adopt a neutral expression. Otherwise, no instructions are given. For example, if all the subjects were asked to hold their lips together or to adopt a blank expression then

some of the natural variation in the way people hold their face would be masked or lost. Noonan syndrome, for example, has an associated slackness of the lower jaw, and subjects often do not naturally hold their mouth closed unless specifically instructed.

3.2 Algorithm

To solve the correspondence problem, we interpolate between a sparse set of hand-placed landmarks. The following sections explain the two steps in the DSM algorithm: surface registration and building a point distribution model.

3.2.1 Manual landmark placement

The first step in the dense surface model algorithm is to place landmarks manually on each surface. For human faces we found that just 9 landmarks gave a model that was visually acceptable (see Fig. 3.14) but, in general, different numbers of landmarks will be needed for different purposes:

- In Chapter 6 we compare the classification ability of models built with different numbers of landmarks. Up to a point more landmarks will improve the correspondence and hence the classification but if landmarks with poor reproducibility are introduced the correspondence could get worse.
- For building combined colour and surface models (Chapter 7) we typically use more landmarks to obtain a better appearance in the texture image.
- More landmarks means longer computation time, so if time is important the number of landmarks could be reduced.
- The number of landmarks required will depend on the shape variation of the object - structures with many components that can vary independently will require more landmarks than structures with fewer components.

Figure 3.2 shows an example mesh with the landmarks overlaid. The landmarks are described in Table 3.1.

Placing the landmarks on the 3D surface can be easily achieved in a graphical user interface by taking the first intersection on the viewing ray from the mouse location to the surface. This operation is directly supported by the visualisation library we are using (VTK). Clicking on the surface in this way is intuitive to the user, and the colour texture, where available, makes it easier to place some landmarks such as along the borders of the lips. With only nine landmarks, each example takes under a minute to annotate by hand. In Chapter 6 we will look at how varying the number of hand-placed landmarks affects the model.

(a) (b)

Figure 3.2: An example scan with our nine landmarks *(a)* and a detail of the mesh around the eye with two landmarks visible *(b)*.

One way of avoiding the manual placement of landmarks is to use some form of automated registration. In Chapter 5 we present a method for fitting that could be used to register new surface scans before adding them to the model.

3.2.2 Forming the dense correspondence

The next step is to establish a dense correspondence between the surface meshes. This could be done using any set of landmarks as a frame of reference but it is desirable that the landmarks are typical of the distribution, so we have used the generalized Procrustes algorithm (Goodall, 1991) to compute the mean landmarks. The key step in this process that is used repeatedly is a least-squares alignment of two sets of 3D landmarks for which we use the quaternion method described for example in Horn (1987).

Each surface is then warped onto these mean landmarks using thin-plate spline (TPS) warping (Bookstein, 1997b). This brings the landmarks into exact alignment and interpolates a smooth transform for the other parts of the mesh, minimizing a bending energy. This is intended to ensure that while all the information implicit in the landmarks is used, as little spurious variation as possible is introduced, especially in the vicinity of each landmark. Thin-plate splines are a particular form of radial basis function (RBF) warping.

Figure 3.3 shows a set of faces aligned in this way, shown from the front and from the side. In the shoulder areas, for example, the surfaces are far apart, these areas are regarded as badly corresponded and are trimmed in the next step. In the face area,

corners of the eyes	Landmarks are placed in the corners of each eye, where the eyelids meet.
corners of the mouth	Landmarks are placed in the corners of the mouth, where the vermillion border stops.
soft tissue nasion	A point on the bridge of the nose, forward of the fronto-nasal suture. The face is typically rotated to check that the placement of this landmark appears correct from different directions.
nose tip	The most forward point on the nose, when the head is judged to be in the natural head posture.
soft tissue gnathion	The most forward and downward point on the chin; the first point of contact with a 45 degree plane when the head is in a natural head posture.

Table 3.1: The protocol used for placing the landmarks on the 3D face surfaces.

however, the scans are close together (the shape of the face can still be seen). These areas are closely corresponded and are kept (see Fig. 3.6).

Figure 3.3: A set of raw surface scans warped using thin-plate splines onto a set of 9 landmarks. Each surface is rendered in a different colour, hence the colour that appears at any point in this picture is the one that is closest to the observer. Note that some of the surfaces include shoulder areas and hair, these will have to be trimmed before building a model.

Having brought all the surfaces into close alignment, the dense correspondence is made by taking the closest point on each surface from each vertex in a base mesh. The connectivity of the base mesh (describing which vertices are joined by triangles) is then applied to the resampled points, giving a set of surfaces all with the same connectivity.

As a base mesh we have typically used one of the examples in the training set. Experiments showed that as long as the area of interest is covered by an adequate triangulation, the choice of which mesh to use is not critical. Figure 3.4 shows the

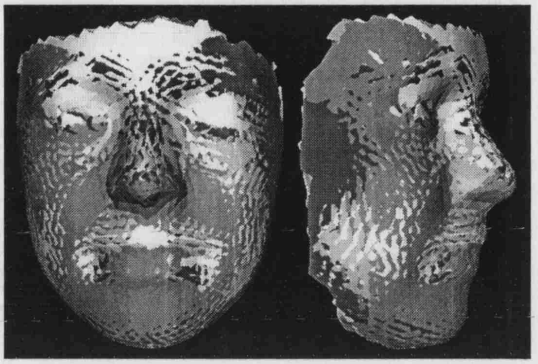


Figure 3.4: The average surface computed using four different base meshes. The mottled appearance shows that these surfaces are near identical.

average surface (see section 3.2.6) computed using four different bases meshes selected at random from the training set. The mottled appearance is due to the fact that the surfaces are very close to each other.

In Lorenz and Krahnstöver (2000), a mesh regularization step is used at this point to get rid of folds and uneven sampling in the surface caused by this use of closest-point mapping. The same problem in addessed in Paulsen and Hilger (2003); Hilger et al. (2004). These sampling problems can be introduced at points of high curvature where the miscorrespondence is large. This effect is illustrated in Fig. 3.5. On a dataset of human faces these effects are typically not seen, as long as at least eight or so landmarks are used, and so we have not found regularization necessary. In any case, these effects should only appear where the correspondence is poor, and if that is the case then post-processing of the mesh to smooth its sampling will not be an ideal solution. By trimming those parts of the surface where the correspondence is poor (see section 3.2.3), we appear to be able to avoid these issues.

3.2.3 Trimming the surfaces

The scans in the training set (including the base mesh) often included significant neck and ear areas that were not present in all the examples. We snip off these areas by using only those vertices where the *maximum* distance from the base mesh to each surface (after alignment) is less than some threshold, k (typically chosen through experiment). While this distance is, of course, application-specific, this technique is very effective in restricting the model to those regions that are well-represented by the training set surfaces. By ensuring that *every* surface has a miscorrespondence of less than k at every point we essentially take an *intersection* of the meshes; keeping only those areas that are well covered in *all* the examples.

Figure 3.6 (left) shows how the maximum miscorrespondence value varies across

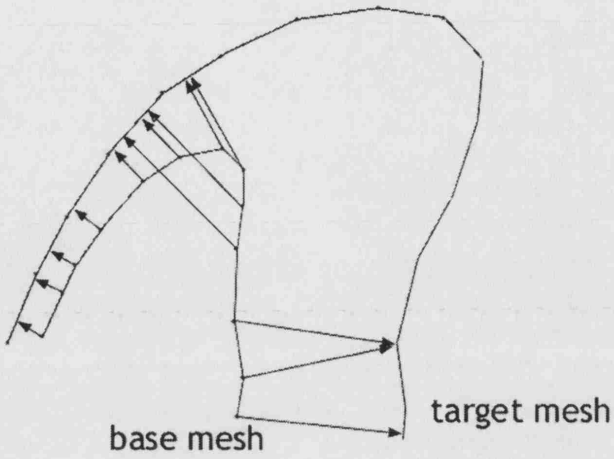


Figure 3.5: Resampling errors can occur where the correspondence between surfaces of high curvature is not very close. The target mesh is resampled by finding the closest point to every vertex in the base mesh. In this rather extreme illustration, the tip of the target surface is not sampled at all, and a fold is introduced into the final surface, where the tip of the base mesh sampled the same bit of target surface twice. Mesh regularization could be used to improve the sampling of the final surface (Lorenz and Krahnstöver, 2000), but a better solution would be to avoid such effects appearing in the first place.

the base mesh for a set of faces. Around the landmarks the miscorrespondence is small, and it grows as you move away. For human faces we often use a threshold of 20mm. Figure 3.6 (right) shows the distribution of maximum miscorrespondence values for the vertices on the base mesh, and the threshold chosen.

3.2.4 Holes and spikes in the input data

In addition to areas of hair and clothing, surface scans acquired through stereo photogrammetry or laser-scanning often have holes, and sometimes spikes, in them. The holes present a problem to algorithms that require the surface to have properties such as being locally manifold at every point, since they must be filled using some form of interpolation. While there are now some sophisticated algorithms for doing this (eg. Carr et al., 2001), manual intervention might be required.

Another issue that can arise is polygons that are not connected to the rest of the surface in a locally-manifold fashion but by only one edge or one vertex. Figure 3.7 illustrates two possiblities. Ensuring that a surface is completely free from such problems is not straightforward in many cases, since it can be hard to decide how the surface should be edited.

The DSM algorithm does not require the surfaces to be cleaned in any way, and

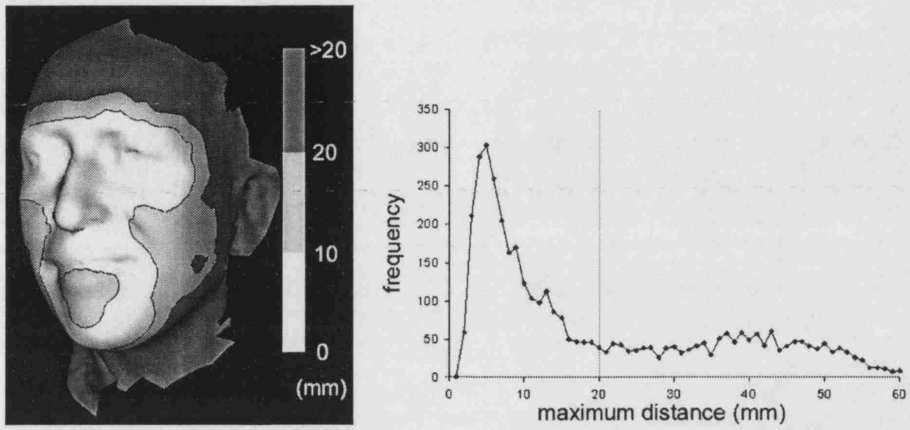


Figure 3.6: The maximum miscorrespondence value changes across the base mesh (left), with low values near the landmarks and larger values as you move away. By imposing a threshold on this value we trim the surface of those parts not well corresponded. On the right is a graph of the distribution of these values, with a threshold of 20mm indicated.

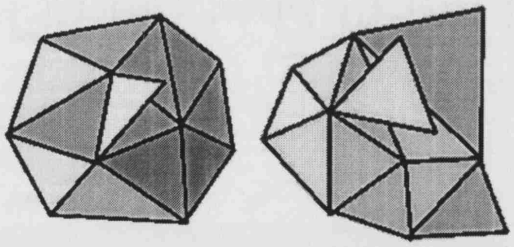


Figure 3.7: Two possiblities for polygons to be non-manifold. On the left a triangle is connected to the surface by only one edge, while on the right by only one vertex. The other (non-boundary) polygons are locally manifold since every edge is shared by exactly two polygons. Such polygons in the surface are problematic for algorithms that require certain properties of the surface.

can build models from surfaces that contain all of the problems discussed above. This is clarified below.

The base mesh is used to resample every surface, by finding the closest points after TPS-warping into alignment. After this resampling step every surface has the same connectivity as the base mesh, only the positions of the vertices vary. Thus all the connectivity problems of the training set meshes will not appear in the final model. Holes and spikes and similar artefacts *do* cause problems in the sampling (Figs. 3.8 and 3.9) but importantly any vertex displacements caused by these issues tend not to be correlated with the major shape changes in the model and are thus only represented in the principal components of very low variance, along with the noise in the model. Since typically we use only the principal components that explain a given percentage of the variation (eg. 98%), any effects introduced by holes or spikes or similar will typically not be seen in the final model.

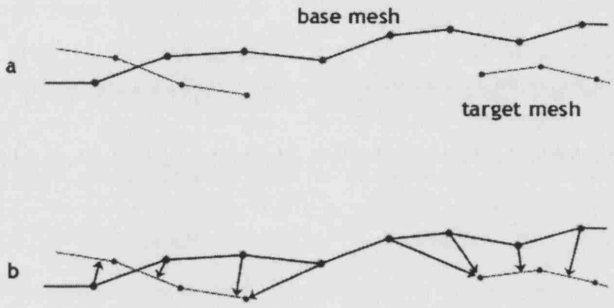


Figure 3.8: The base mesh and target mesh illustrated in 2D (a), with a hole in the target mesh. In such situations the holes cause the base mesh to sample at the edges (b). If the hole is very large then many vertices will be displaced but small holes will displace only a few vertices and the effect on the model will be minimal, since the displacements will not be correlated with the major shape changes.

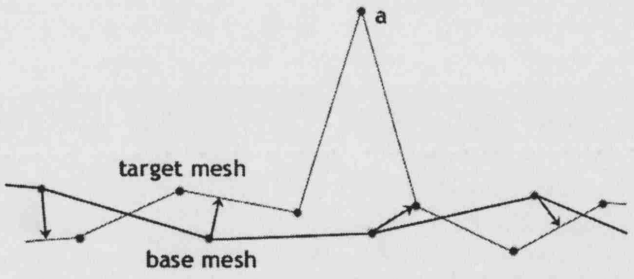


Figure 3.9: In a similar fashion to holes, spikes (a) in the training set surfaces will typically not cause problems since the base mesh will sample around them. Again, unless the spike involves many vertices the effects on the model will be minimal.

3.2.5 Unwarping the resampled surfaces

After TPS-warping each surface to the mean landmarks and then resampling them using the base mesh, the next step is to return the resampled surfaces to their original position, by 'unwarping' them. To do this, we find the barycentric coordinates of the resampled vertex within the triangle on the warped mesh that contained it, and move the vertex to the same relative position in the corresponding triangle in the original surface. Figure 3.10 illustrates all three steps.

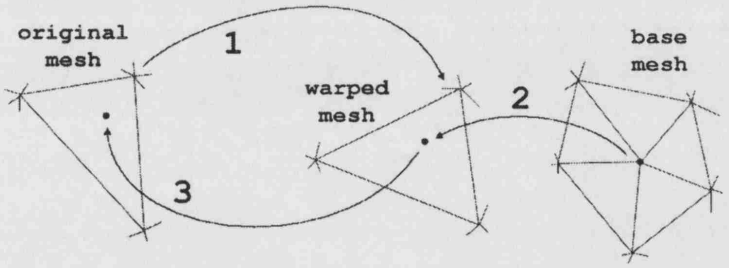


Figure 3.10: The dense correspondence algorithm illustrated. Every mesh (including the base mesh) is first TPS-warped into alignment with the mean landmarks (1). Each warped mesh is then resampled by finding the nearest point to each vertex of the warped base mesh (2). Finally the resampled vertex is returned to the same relative position within the corresponding triangle on the original mesh (3).

The result of all these steps is a set of trimmed and resampled surfaces lying close to their original positions, and having close to their original shape. Each vertex in the resampled surface is guaranteed to lie on the original surface. For a visual demonstration of the correctness of the correspondence, see Fig. 7.8 (p. 101).

3.2.6 Building the point distribution model

Now that we have constructed corresponding vertices in all the surfaces, we can treat them as landmarks. Following Cootes et al. (1995), we first apply the Procrustes algorithm to align all the shapes and produce a mean shape. Because our data is calibrated for size, we do not include scaling in the Procrustes alignment, but instead build a size-and-shape model (Dryden and Mardia, 1998).

Figure 3.11 shows the age distribution for the dataset we will be using in the next few sections. 193 scans of different people are used, with an approximate balance of male and female at the different ages. In total there are 82 females and 111 males. The youungest subject is three months old, while the oldest is 76 years. 27 of the subjects have Noonan Syndrome, while the majority (166) have no syndrome.

Figure 3.12 shows the mean mesh that was computed for a dataset of 193 faces.

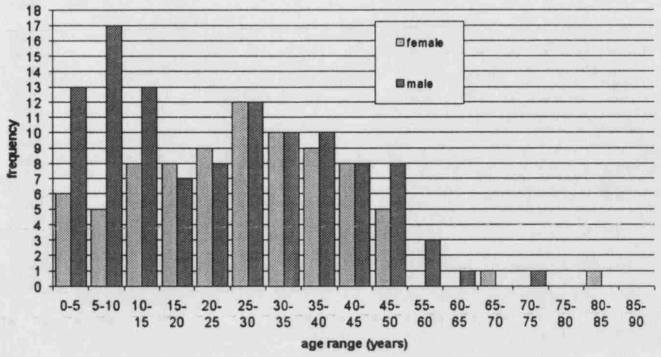


Figure 3.11: The age distribution, by gender, for the dataset of 193 faces.

A visual check that this mesh is smooth and free of artefacts such as vertex bunching gives us some indication that our landmarks are sufficient and placed correctly. The mesh has been clipped of the poorly represented areas, leaving it with 1688 vertices in this case.

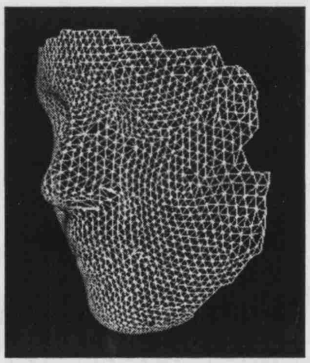


Figure 3.12: The averaged mesh, restricted to only those vertices that had good correspondences in the training set.

The next step is to apply a principal components analysis (PCA) to the data. Each example can be represented by a shape vector of concatenated x, y and z coordinates:

$$\mathbf{x}_i = [x_1, y_1, z_1, \ldots, x_n, y_n, z_n]^T \tag{3.1}$$

The mean shape vector is given by:

$$\overline{\mathbf{x}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{x}_i \tag{3.2}$$

where s is the number of examples in the training set (193 in this example).

We compute the $3n \times s$ matrix $\mathbf{D}$ using:

$$\mathbf{D} = [(\mathbf{x}_1 - \overline{\mathbf{x}})|\ldots|(\mathbf{x}_s - \overline{\mathbf{x}})] \tag{3.3}$$

The covariance matrix $\mathbf{S}$ can then be computed using:

$$\mathbf{S} = \frac{1}{s-1}\mathbf{D}\mathbf{D}^T \tag{3.4}$$

$\mathbf{S}$ is eigen-decomposed to give a set of eigenvectors ϕ_i and eigenvalues λ_i. However, because there are n vertices in the mesh, each shape vector $\mathbf{x}_i$ has $3n$ elements (5064 in this case), so $\mathbf{S}$ has $(3n)^2$ elements. Fortunately, we can avoid having to eigen-decompose the very large matrix $\mathbf{S}$ by instead making use of the Eckart-Young theorem (Johnson, 1963) and computing the $s \times s$ matrix $\mathbf{T}$:

$$\mathbf{T} = \frac{1}{s-1}\mathbf{D}^T\mathbf{D} \tag{3.5}$$

which is somewhat more manageable. It follows from Johnson (1963) that the first s eigenvalues of $\mathbf{S}$ are the same as those of $\mathbf{T}$, the remainder are zero. We can compute the first s eigenvectors of $\mathbf{S}$ from the eigenvectors $\mathbf{e}_i$ of $\mathbf{T}$ using (again from Johnson, 1963):

$$\phi_i = \mathbf{D}\mathbf{e}_i \tag{3.6}$$

The eigenvectors ϕ_i must be renormalized (scaled to unit length) after this step.

3.2.7 Deforming the shape template using parameters

The computed eigenvectors can be treated as deformations of the whole mesh and can be directly added to the coordinates of the vertices of the mean mesh to synthesise new surfaces:

$$\mathbf{x}_{\text{new}} = \overline{\mathbf{x}} + \Phi\mathbf{b} \tag{3.7}$$

where $\Phi = [\phi_1|\phi_2|\ldots|\phi_t]$ is the matrix of the first t eigenvectors and $\mathbf{b} = [b_1, b_2 \ldots b_t]$ is a set of parameters controlling the modes of shape variation.

The shape modes are not of equal importance, instead the majority of the shape variation in the training set is typically described by the first few components. The lower components describe the uncorrelated shape variation, typically the noise in the positions of the vertices. For speed of synthesis therefore, not all of the components need be used to obtain an accurate synthesis of a given shape (see Ch. 4 for more on this). The value of t is typically determined by the number of components that are required to account for a certain percentage of the variation, often 98%, ie. the lowest

value of t such that:

$$\frac{\sum_{i=1}^{t} \lambda_i}{\sum_{j=1}^{s} \lambda_j} \geq 0.98 \tag{3.8}$$

3.2.8 Saving and loading the model

The advantage of restricting t in this way is that applying PCA allows a lot of redundancy to be removed, since the correlations between the coordinates of the vertices are described in the principal components. This results in a compact model, saving not only on rendering time but also on storage requirements. To store a dense surface model we write the items listed in Table 3.2 to disk.

eigentotal	the sum of all the eigenvalues
t	the number of eigenvalues saved (since we need not store all of them)
eigenvalues	the first t eigenvalues, in descending order of size
n	the number of vertices in the mesh
eigenvectors	the $t \times 3n$ matrix of eigenvectors (order matching the eigenvalues)
tcoords	the texture coordinates (t_x, t_y) for each of the n vertices
m	the number of polygons in the mesh
polygons	m tuples of indices into the list of vertices
mean surface	the coordinates of the average surface vertices
l	the number of hand-placed landmarks used to build the model
mean landmarks	the coordinates of the mean landmarks

Table 3.2: The items that are stored to disk when saving a dense surface model. The landmarks are required when synthesising new examples, since they must be TPS-warped into correspondence before being resampled and projected into the shape-space.

The eigentotal is saved in order to be able to compute what proportion of the overall shape variation each mode accounts for. For example, if the eigentotal is 25653.01 and the first eigenvalue is 20629.66, then the first PC accounts for 80.42% of the variation. The percentages themselves could be stored instead but the eigenvalues are useful also for when two PCA models need to be combined, as discussed in Chapter 7.

For an explanation of *tcoords* and *polygons*, see Chapter 7 where representations of textured polygonal surfaces are discussed.

The mean landmarks are required in order to be able to represent new surfaces in the model. This is discussed in the next section. One complication is that the mean landmarks that are stored are *not* the same as the ones originally computed (section 3.2.2), since these do not necessarily lie on the mean surface as computed in section 3.2.6. Instead, we take the transform computed for each surface when it was being Procrustes-aligned (section 3.2.6), and apply this to the set of landmarks for

that surface. The mean landmarks are then recomputed. This ensures that the mean landmarks lie on the mean surface, and simplifies the steps necessary to resample a new surface, as discussed in the next section.

3.2.9 Synthesising new surfaces

To find where a new surface scan lies in shape-space relative to the other examples in the model, we need to work out what parameters $\mathbf{b}$ best represent it. Before we can compute $\mathbf{b}$ we need to resample the surface using the same set of vertices used in the model. There are two methods for doing this. Either we can automatically fit the model to the new example (see Chapter 5 for discussion of this), or we can use a set of landmarks on the surface to register it.

Since in the model we store the coordinates of the mean landmarks in the model, we can TPS-warp the surface to be synthesised onto the mean surface. This allows us to resample the target surface as usual (Fig. 3.10), by finding the closest points to the vertices of the mean surface and then unwarping the surface using barycentric coordinates. This gives us the shape vector $\mathbf{x}$.

To find the parameters $\mathbf{b}$ necessary to synthesise the new face $\mathbf{x}$ we use (following Cootes et al., 1995):

$$\mathbf{b} = \Phi^{\mathrm{T}}(\mathbf{x} - \bar{\mathbf{x}}) \tag{3.9}$$

The parameters b_i are in 'raw' units - those of the input space. To convert to 'standardised units' (ie. standard deviations) and back is straightforward, using:

$$b'_i = \frac{b_i}{\sqrt{\lambda_i}} \tag{3.10}$$

We represent the parameter vector $\mathbf{b}$ as $\mathbf{b}'$ when expressed in units of standard deviation.

3.2.10 Modelling the population

After making the correspondence the surfaces in the training set all lie in a *shape-space*, the space spanned by their shape vectors. With many vertices in the surfaces, this space is of high dimensionality. For example, with $n \approx 2000$ vertices in the 3D mesh, this space has 6000 dimensions. Of course there are only $s = 193$ examples, so there are insufficient points to span the entire space. Just as three points always lie in a two-dimensional plane, so s points span $s - 1$ dimensions. Within these $s - 1$ dimensions, the scatter of points lie in some distribution, and applying PCA tells us in which directions there is most variation and in which directions there is least.

The shape of the distribution in this space is the key issue when trying to model

the population. If the distribution is a Gaussian scatter around some central point then we can construct a very simple model, using the principal components to find the hyper-ellipsoid that should contain 98% of the population.

If the distribution is not Gaussian then some other model must be used. Possibilities include:

- mixture of Gaussians (eg. Cootes and Taylor, 1997),
- Independent Component Analysis (ICA) (eg. Üzümcü et al., 2003),
- non-linear PCA (eg. Sozou et al., 1997),
- kernel-PCA (Schölkopf et al., 1998).

However the simplest model we can assume is the Gaussian, and so this becomes our default choice. We model the distribution using the assumption that as long as the synthesised example has a Mahalanobis distance of less than three standard deviations from the mean then it will be within the variation seen in the training set and will be a plausible human face. With the parameter vector $\mathbf{b}'$ expressed in standard deviations, the Mahalanobis distance, r, and the Euclidean distance are equivalent, being given by:

$$r = \|\mathbf{b}'\| = \sqrt{\mathbf{b}' \cdot \mathbf{b}'} \tag{3.11}$$

Deciding whether a distribution is well-modelled by a Gaussian is a non-trivial task. Scatter plots between pairs of modes such as that shown in Fig. 3.13 can be examined to check that the distribution contains no obvious non-linearities, such as banana-shaped curves, separate groups or skewing. This scatter plot shows no obvious deviations from a Gaussian distribution, although we suspect more examples would show skewing to one side.

3.3 Results

Figure 3.14 shows the first three principal components (modes) that were computed for the 193 examples. These were created by using

$$\mathbf{x}' = \overline{\mathbf{x}} + k\sqrt{\lambda_i}\phi_i \tag{3.12}$$

where $k \in \{-3, 0, +3\}$ and $i \in \{1, 2, 3\}$. As in section 3.2.6, λ_i are the eigenvalues and ϕ_i are the eigenvectors.

This figure gives visual confirmation that the DSM algorithm seems to be working as desired - all the synthesised faces in Fig. 3.14 are plausible. Since these faces come

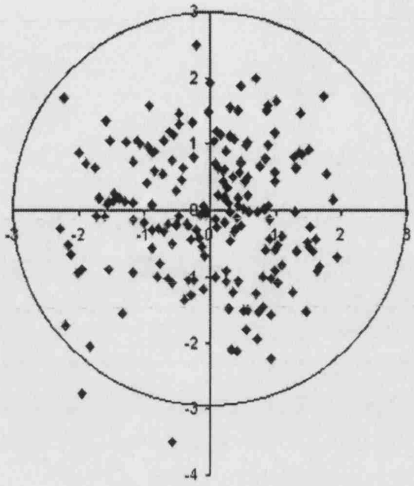


Figure 3.13: A scatter plot of mode 1 (horizontal) against mode 2 (vertical), showing a reasonably Gaussian distribution. Units are standard deviations. The 3 standard deviation boundary is shown as a circle.

from the boundaries of the region in face-shape-space that was identified as being valid, the fact that they look like human faces is highly encouraging.

Also, the synthesised faces in Fig. 3.14 do not contain any of the holes or surface errors that were present in the training set, since we chose a base mesh that was free of such problems. Any noise present in the model is not likely to be correlated with the major changes in shape being captured by the principal components and thus will only appear in the lower modes. This feature of the surface is highly beneficial when working with real-world data, any method that relied on the input being manifold throughout would be less useful.

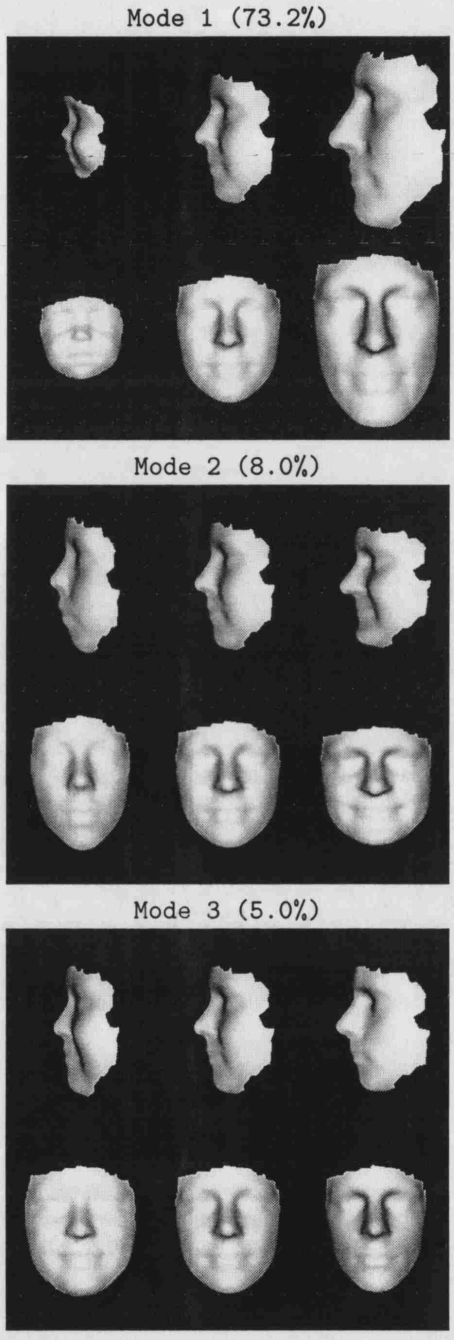


Figure 3.14: The first three modes, between -3, 0 and +3 standard deviations (columns), side and front views (rows).

3.4 Conclusions

The prime motivation for the development of the DSM algorithm was to make use of the 'extra data' present in a surface scan. The traditional approach for analysing the shapes of a class of three-dimensional objects - 'geometric morphometrics' - is to place landmarks across each surface at recognisable points. For a face this would mean marking points such as the corners of the eyes, plus points on the nose, mouth, chin and ears. For some landmarks there is the concept of a *biological homology* (MacLeod, 2001). Such landmarks are points that correspond anatomically between the subjects. Examples include fissures on the skull and well-defined soft-tissue points such as the inner corners of the eyes. Other landmarks cannot be located so precisely. Some, like the tip of the nose, are dependent upon the precise orientation of the head. Some landmarks can only be located accurately in one dimension, such as points along the border of the lower lip. These are sometimes called *semi-landmarks* (Bookstein, 1997a). The number of reliable landmarks that can be placed is invariably far fewer than the thousands or tens of thousands of points on laser-scanned or photogrammetry-acquired surfaces. With DSMs we can make use of the additional shape information present in the entire surface.

A major application of this extra data is to improve classification, this is explored in Chapter 6. In the next chapter we will evaluate the technique by looking at how well models generalise, by testing their sufficiency and specifity. Later chapters evaluate applications of the technique.

Chapter 4

Modelling Capacity of DSMs

In this chapter we explore some of the properties of DSMs as applied to real data. Later chapters evaluate the models in more detail.

4.1 Introduction

Having shown in a preliminary fashion that the model generates valid faces, it is worth checking in more detail that other properties hold true. Any model of a population must satisfy two basic properties: sufficiency and specificity. That is, it must model valid examples reasonably accurately and it must *not* model invalid examples (examples not in the population being modelled). By using subsamples of the data available we can obtain an estimate of how well the wider population can be modelled, although this of course assumes that our dataset is representative of the whole population. In general, any test of sufficiency and specificity will be valid only for the dataset on which it is run.

We have two handles on the sufficiency of the model: how well it can model the examples in the training set and how well it can model examples not in the training set but known to be of the same type.

For the specificity of the model we need to test whether the model is capable of synthesising objects which are not valid faces; examples that were not drawn from the same population as our training set. The specificity of the model is evaluated in section 4.3.

4.2 Sufficiency

We built a DSM using the same data set as in the previous chapter: 193 scans of different peoples' faces. This gives for every training set surface (A) a resampled and trimmed surface (B) that has different vertices and triangles but almost exactly the same shape in the face area of interest. In addition we can synthesise surface B using different numbers of modes, t, giving an approximation (C).

To compute how well the synthesised version, C, approximates B we do the following. For each resampled mesh B in the training set:

1. we computed the number of modes p required to model a given percentage of the variation,
2. we computed the parameters $\mathbf{b} = \{b_1, b_2, \ldots b_p\}$ required to best model shape B (using equation (3.9)),
3. we computed the shape C given by the parameters $\mathbf{b}$,
4. we computed the Euclidean distance between each pair of corresponding vertices in the resampled surface B and the synthesised surface C, and
5. we computed the RMS, mean and maximum distance across the set of vertices in the mesh.

The RMS, mean and maximum distances were then averaged over the training set. The results are presented in Table 4.1.

%	modes req'd	RMS (mm)	mean (mm)	maximum (mm)
95.0	13	1.57	1.31	7.70
98.0	32	0.99	0.81	6.11
99.0	56	0.71	0.58	4.41
99.9	146	0.22	0.18	0.89
100.0	191	0.00	0.00	0.00

Table 4.1: In-training-set synthesis errors. This table shows, for different proportions of the variation being modelled, the number of modes this requires and the RMS, average and maximum vertex errors for synthesising faces from a training set of 193 examples. Here we are comparing the synthesised face with the resampled version of the original surface scan.

As expected, the errors decrease as the percentage of variation modelled increases, falling to zero when all the modes are used. Typically 98% is used in the literature, at which level we see a mean and RMS vertex error of under 1mm.

This comparison was with a resampled version of each shape, not the original. This resampling step is itself a source of error and so we need to incorporate it into the experiment. To do this we compare C with A, rather than C with B.

Since surfaces C and A have a different number of vertices we cannot directly compute an error on each vertex. Instead we have to find the closest point on surface A to each vertex in the mesh C and use this distance. Again we find the RMS, mean and maximum distance for the vertices in each mesh C and then average these figures over the entire training set. The results are given in Table 4.2.

%	modes req'd	RMS (mm)	average (mm)	maximum (mm)
95.0	13	1.40	1.07	7.02
98.0	32	1.16	0.87	6.05
99.0	56	1.03	0.77	4.97
99.9	146	0.90	0.67	3.35
100.0	191	0.88	0.66	3.26

Table 4.2: In-training-set synthesis errors. For different proportions of variation, this table shows the number of modes required and the RMS, average and maximum vertex errors for synthesising faces from a training set of 193 examples. Here we are comparing the synthesised face directly with the original surface scan.

The figures in Table 4.2 are a more realistic estimation of the (in-training-set) capabilities of the model, since they reflect the necessary resampling of the surface. The errors do not drop to zero since the variation in the extent of each input and the holes they contain are not being modelled. Computing the error values in Table 4.2 allows us to compare these in-training-set results with the out-of-training-set results.

To explore more fully the model sufficiency we need to test it on unseen data. We can do this by randomly subsampling the training set, rebuilding the model from the subsample, and using part of the remainder as a test set. The experiment shows us how the errors change as the size of the training set is varied, keeping the amount of variation modelled at 98%. To avoid the pitfall of choosing a training set that just happens to be particularly good or bad we average the performance over repeated runs.

With enough data we should expect to be able to model unseen surfaces as well as the ones in the training set, the question is how much data we would need to do this. Table 4.2 gives us the target for which we are aiming - for a specific percentage of the modes (98%) with enough data we would expect to see the same vertex errors.

We use the same base mesh throughout. The steps in the experiment are as follows:

1. randomly select with removal n examples from the entire dataset to form a training set, and build a DSM model from this data,

2. from the remainder, randomly select with removal m examples to serve as a test

set,

3. TPS warp each test example A from its landmarks to the mean surface landmarks, resample using the mean shape and unwarp, this gives shape B (this step is the core of the DSM algorithm),
4. compute the number of parameters p required to model 98% of the variation,
5. compute the p parameters required to model shape B,
6. compute the shape C modelled by the parameters,
7. compute the closest-point distance between each vertex in the mesh C and the original test surface A,
8. repeat from 3. for each surface in the test set using this model, and
9. repeat from 1. k times for each size of training set

The size of the training set, n, was varied between zero and the number of examples available (193) minus m. The parameters m and k were both kept fixed at 10 for this experiment.

For each test surface the RMS, mean and maximum vertex errors were calculated. These figures were then averaged over the test set (10 examples) and *these* figures were then averaged over the 10 training set samples. Thus the values in Table 4.3 are each averages of 100 test surfaces. (In the last run, the program ran out of memory, and so only 8 samples could be taken.)

training set	modes	vertices	RMS (mm)	mean (mm)	max (mm)	samples
19	12.5	2163	2.44	1.76	13.6	10
38	19.9	2026	1.76	1.29	10.5	10
57	24.5	1955	1.67	1.22	9.78	10
76	26.5	1885	1.54	1.14	9.07	10
95	28.4	1845	1.51	1.13	8.71	10
114	29.2	1806	1.35	0.99	7.82	10
133	30.3	1791	1.32	0.98	7.24	10
152	31.7	1802	1.36	1.03	7.25	10
171	30.5	1730	1.27	0.94	7.19	8

Table 4.3: For different sizes of training set, this table shows the average number of modes required to model 98% of the variation, the average number of vertices in the mesh, and the RMS, mean and maximum vertex errors. These figures are displayed graphically in Figs. 4.1, 4.2 and 4.3.

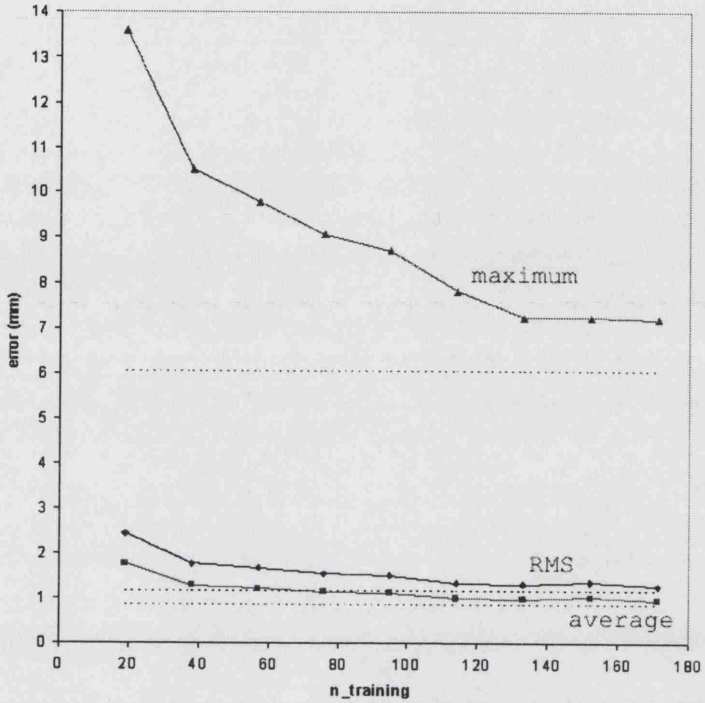


Figure 4.1: The maximum, RMS and mean vertex errors for different tests runs with increasing training set size. The target in-training-set errors (from the 98% row in Table 4.2) are shown as dotted lines.

In theory, with a large enough training set we should be able to model unseen examples as well as we can model examples in the training set, since they come from the same population. Figure 4.1 shows that the data matches this expectation; the errors approach their in-training-set values (shown as dotted lines) as more examples are added. The graph tells us that, while a larger training set will always give better results, with approximately 150 faces our model will have enough data to perform well.

This result is confirmed in Fig. 4.2, since the 98% line levels out when the size of the training set has grown to 100 examples or so. This tells us that the majority of the shape variation of human faces is being captured by approximately 30 modes which can be extracted from 100 faces. The only beneficial effect of adding more examples beyond this point is that the other 2% of the population may be better represented.

Figure 4.3 shows how the number of vertices in the modelled surface drops gradually as more examples are added. This is because the mesh is trimmed to only those areas where the maximum distance across the corresponded training set is less than 20mm. Occasionally a new example will have a large hole or gross errors in a region of the face where none was previously encountered, causing this part of the face to be snipped off.

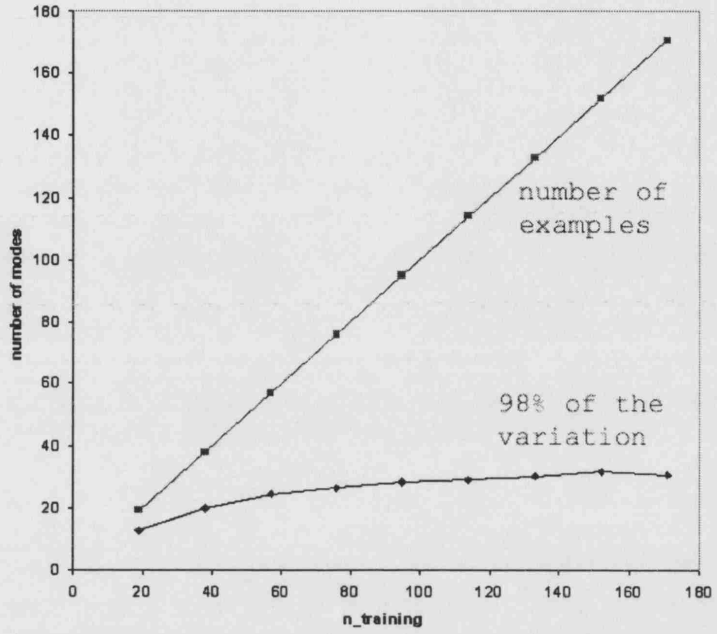


Figure 4.2: The number of modes required to model 98% of the variation does not increase linearly with the size of the training set.

For a given acquisition system such as the DSP400 using subjects with little or no facial hair we would expect the number of vertices in the modelled region to stabilise since the captured areas are very similar. If a surface with large holes (caused for example by the subject having a lot of bushy facial hair) were then included in the data set, the modelled region could shrink further.

4.3 Specificity

Having shown that the DSM model is capable of synthesising examples that are valid, we also need to check that it is *not* capable of synthesising examples that are *not* valid. Here we use valid to mean plausibly a member of the same population from which the training set was drawn, which for the moment means non-syndromic human faces.

It is not possible to prove the specificity of the model since there is an infinite range of non-faces. However, we can illustrate specificity on a number of test cases and at least convince ourselves that the model performs as hoped on these. We can look at specificity in two ways. Firstly, if we find surfaces within the region of shape-space defined by the training set that do not look like human faces then we would have to say that the model is not specific. Our first experiment searches for these *false positives* by

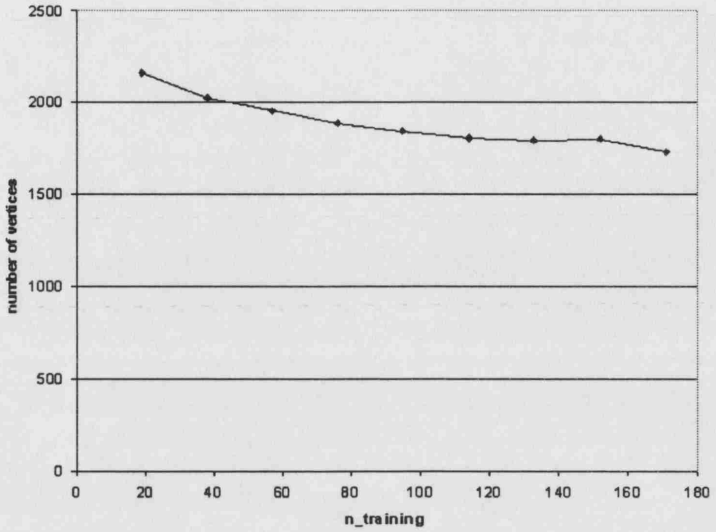


Figure 4.3: The number of vertices decreases slightly as more examples are used.

sampling the shape-space.

Secondly, we can try to synthesise objects that are not valid faces and show that their location in shape-space puts them well outside of the region defined by the training set. Our second experiment demonstrates a few of these *true negatives.*

4.3.1 Searching for false positives

We use the same DSM model as before, built from 192 face scans of different people. To model 98% of the variation, $t = 41$ mode parameters were required. In a t dimensional Normal distribution, the χ^2 critical values give us the size of the ellipsoid that we need to include a given percentage of the cases. For 98% of cases in 41 dimensions we need the 8SD ellipsoid. Figure 4.4 shows synthesised faces generated at random, each with:

$$||\mathbf{b}'|| = 8 \tag{4.1}$$

or equivalently:

$$\sqrt{\sum_{i=1}^{t} \frac{b_i^2}{\lambda_i}} = 8 \tag{4.2}$$

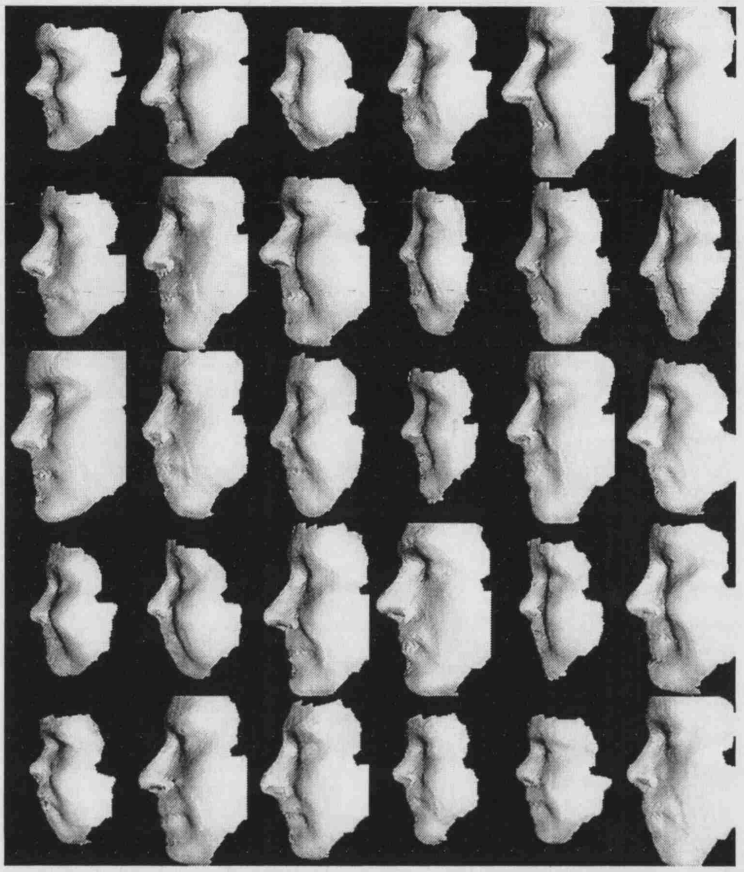


Figure 4.4: Faces selected at random from around the extremes of the 98% ellipse, from a training set of 193 examples.

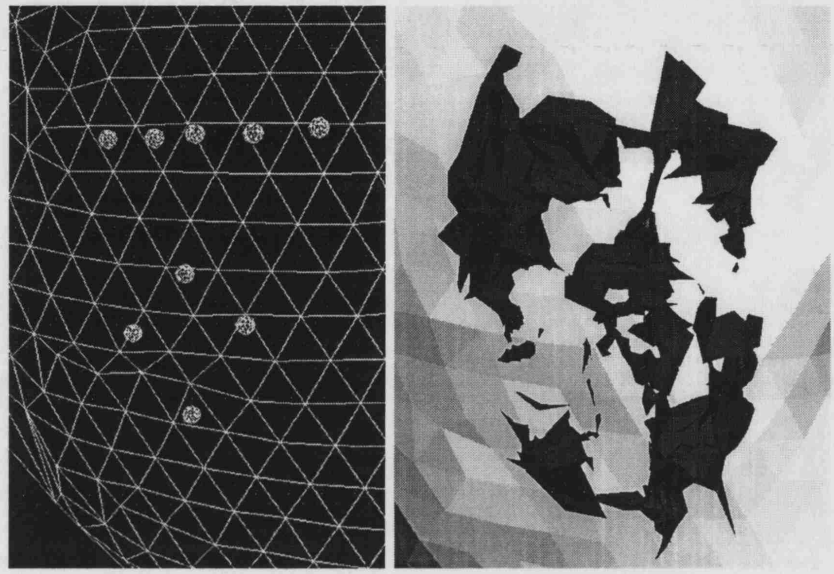
The faces in Fig. 4.4 are all plausible faces, and exhibit a wide degree of shape variation, telling us that the demarcation of shape-space by the 98% ellipsoid appears to be valid.

4.3.2 Searching for true negatives

To synthesise non-face objects we need to be able to project them into the shape-space somehow, in order to examine where they lie in relation to the legal region we have defined. The procedure for doing this is the same as used in section 4.2, ie. we need to resample the objects using the base mesh and then compute the mode parameters using equation 3.9. The resampling step requires hand-placed landmarks on the object. This is a bit odd since we are putting face landmarks on an object that is not a face but it serves to illustrate the performance of the model under such conditions. (An alternative

approach would be to automatically fit the model to the surface, see chapter 5.)

Figure 4.5(a) shows a non-face object (a gently curving piece of surface, shown in wireframe) with our 9 hand-placed landmarks, while 4.5(b) shows the synthesised version (in dark, with the target surface in a lighter shade). Clearly the result does not look anything like a face, and indeed $||\mathbf{b}'|| = 18.95$. This is significantly outside the 8SD ellipsoid region.



(a) (b)

Figure 4.5: (a) A non-face object (shown in wireframe) with nine face landmarks and (b) its synthesis from the model (in a dark shade), shown on top of the original surface (in a lighter shade). Clearly this surface is not modelled well since the resulting synthesis does not look like a face, and the magnitude of its parameter vector (18.95 standard deviations) tells us that it lies outside of our region of shape-space that we have defined as valid for this dataset (the 8SD ellipsoid).

4.4 Conclusions

In this chapter we have examined some of the generalisation abilities of the model. By looking at how well the model can synthesise unseen examples we can estimate the number of training set surfaces needed to model a population to within whatever accuracy is required. In the next chapters the generalisation ability of the model is used for automatically registering new surface scans (Chapter 5) and for classifying new scans (Chapter 6).

Chapter 5

DSMs for Fitting

This chapter evaluates DSMs as a method for automatically registering new examples, by fitting a deformable template to them. In later chapters we look at other uses of DSMs.

5.1 Introduction

The manual annotation of surfaces can be a time-consuming process and subject to errors. If 3D scanning is ever to be used for the wide-scale detection of syndromes, for example, as part of a screening process, then we ideally need some method of automatically interpreting a newly acquired image. Other applications would include identity verification for security purposes, tracking of dynamic sequences for expression analysis and (facial) feature finding for integrating 3D scans into games or for other manipulations.

One of the important uses of point distribution models (Cootes et al., 1992) was for fitting to images, as active shape models (Hill et al., 1992; Cootes et al., 1995). In this chapter we show how DSMs, being an extension of PDMs to 3D surfaces, can also be used for fitting. As with ASMs, the key insight is that we can constrain the template to represent only 'legal' examples, preventing the surface from deforming in ways that do not look like the examples in the training set.

The issue of where to initialise the template before fitting is as problematic as with ASMs. The deformable template is quite prone to becoming stuck in local minima; where there is structure in the scene other than the target the template can latch onto this and not converge correctly. We briefly discuss below how this situation might be detected and avoided.

For our evaluation of the fitting we will measure the final error after starting the template at varying offsets from the correct position. This lets us see that the template

does indeed converge from a range of starting positions and also gives us a handle on the robustness of the convergence. The technique was used for example in Fitzgibbon (2001) to illustrate the *convergence basins*, the range of a parameter over which the process converges.

5.2 Background

As mentioned, the starting point of our method is a combination of ASMs and ICP, since we are fitting a deformable model to a polygonal surface.

Since ICP was first proposed there have been many variants suggested that are of interest here. Some, such as the use of space-partitioning to speed up the search for closest points (Zhang, 1994) are incorporated in our method through our use of the VTK library (Schroeder et al., 1997). More importantly, using *robust statistics* (Huber, 1981) for ICP (Zhang, 1992; Fitzgibbon, 2001) can lead to wider convergence basins since the fitting becomes less sensitive to errors. With the traditional least-squares implementation of ICP, points that are not closely matched are given a disproportionately large weight, an unfortunate occurrence because they are likely to be outliers (mis-correspondences because of other structures in the target). By using instead a weighting function that does not increase dramatically as the error increases, the robustness of the ICP algorithm can be improved (Fitzgibbon, 2001).

Figure 5.1 shows two kernels that have the desired properties, the *Lorentzian*, given by:

$$\epsilon(r) = \log(1 + \frac{r^2}{\sigma}) \tag{5.1}$$

and the *Huber* kernel, given by:

$$\epsilon(r) = \begin{cases} r^2 & r < \sigma \\ 2\sigma|r| - \sigma^2 & \text{otherwise} \end{cases} \tag{5.2}$$

where σ is a parameter controlling the roundedness of the central region in Fig. 5.1 and r is the input value.

The Huber kernel in particular is a direct solution to the problem, being essentially a rounded-off (hence differentiable) version of the absolute function ($y = |x|$).

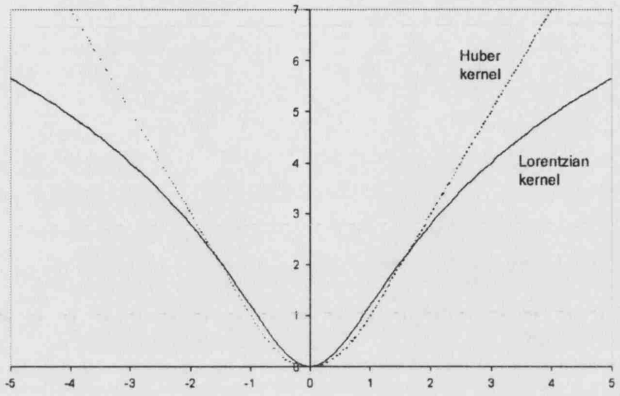


Figure 5.1: The Lorentzian and Huber kernels, shown for $\sigma = 1$. The Lorentzian is scaled vertically by a factor of 4 to emphasise the similarity of the two kernels.

The use of the Huber kernel in ICP was made possible by the insight that using a general-purpose optimisation routine such as Levenberg-Marquardt (Marquardt, 1963) need not necessarily be slower than the purpose-built method at the core of ICP. This is perhaps counter-intuitive since, because rigid-body fitting requires search in a six-dimensional space, it would normally be expected to be slower than the directed search offered by ICP. Part of the solution is that by explicitly minimising the error function it becomes possible to precompute both the distance map and the derivatives of the distance map that are needed in the optimisation (Fitzgibbon, 2001).

Intriguingly, the core of the ASM algorithm is also a directed-search solution to the high-dimensionality problem. In addition to the six pose parameters we need to fit an additional t shape parameters, often of the order of 40 or 50 in total for 3D face surface data. Even without the efficiency improvements of precomputing the distance maps it may be that using robust statistics can give significant improvements to the width of the convergence basins. Some recent work on using robust statistics for ASM fitting seems to show that this is the case (Rogers and Graham, 2002).

5.3 Data

In this chapter and in some later chapters we are using a larger dataset of 421 scans. Of these, 208 are female, 213 male. 324 of them have no syndrome, while 83 have Noonan Syndrome and 14 have Velo-cardio-facial Syndrome. The genders are equally spread out across the age range, as shown in Fig. 5.2.

21 of these scans were selected at random to be the test set, leaving the remaining 400 faces as the training set.

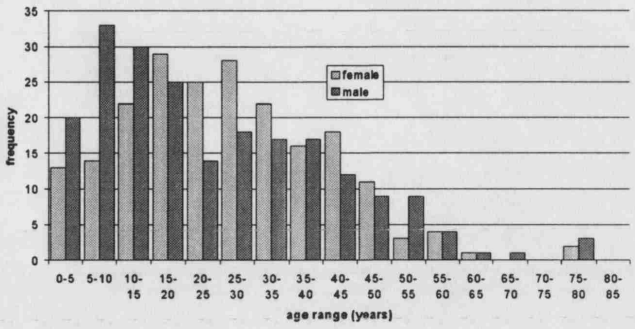


Figure 5.2: The age and gender distribution for the dataset of 421 different faces.

5.4 Method

It would be interesting to apply some of the techniques described above to the DSM fitting procedure, and certainly this should become part of the future work. However, the method described here is an extension to 3D of active shape models (ASMs). While in 2D there are four pose parameters (x and y position, rotation and scale) that must be adjusted in addition to the t shape parameters, in 3D there are potentially seven pose parameters (xyz position, xyz rotation and scale) that must be optimized. Our method for fitting in 3D is a hybrid of ICP (Besl and McKay, 1992) and ASM search (Cootes et al., 1995), since we are not searching for grey-level profile matches (as would be the case if we were fitting to MRI or CT volume images) but instead for polygonal surfaces.

As with ASMs, the search is initiated by placing the mean surface into the scene and iteratively deforming it using the shape parameters $\mathbf{b}$ to best match what is found locally around each vertex. With 2D and 3D images, a model of the target grey-level profile or appearance must be built in order to drive the search but with surfaces the task is much easier, needing only a simple nearest point search.

The key insight of shape model fitting is that the deformable template can be constrained only to vary within a range of shapes defined by the training set. This is step 7 in the sequence of operations below:

1. the initial template is $\mathbf{x}(0)$
2. the target is $\mathbf{y}$
3. standard ICP procedure (Besl and McKay, 1992) is used to fit the template $\mathbf{x}(0)$ to the surface $\mathbf{y}$, giving $\mathbf{x}(1)$
4. closest-point mapping (see below) $\mathbf{x}(1)$ onto $\mathbf{y}$ gives $\mathbf{x}(2)$
5. $\mathbf{x}(2)$ is least-squares (LS) aligned (initially using the Euclidean transform group) with $\overline{\mathbf{x}}$ to give $\mathbf{x}(3)$
6. the t parameters required to model $\mathbf{x}(3)$ are computed:

$$\mathbf{b} = \mathbf{W}^{-1}\Phi^{\mathrm{T}}(\mathbf{x}(3) - \overline{\mathbf{x}})$$

7. the parameters $\mathbf{b}$ are limited to be legal in some way (see below), giving $\mathbf{b}'$
8. the best-guess surface is therefore computed: $\mathbf{x}(4) = \overline{\mathbf{x}} + \Phi\mathbf{W}\mathbf{b}'$
9. $\mathbf{x}(4)$ is LS-aligned with $\mathbf{x}(2)$ to give a new template $\mathbf{x}(5)$
10. if the RMS vertex difference between $\mathbf{x}(1)$ and $\mathbf{x}(5)$ is greater than some threshold ϵ then $\mathbf{x}(1) \leftarrow \mathbf{x}(5)$ (ie. the new template is used for the next iteration) and repeat from step 4
11. otherwise, repeat from step 4 but extend the transform group for the LS-alignment in steps 5 and 9 first to the similarity group and then to the affine group.

By "closest-point mapping" in step 4 we mean moving each vertex in the template to the closest point on the target surface, not necessarily itself a vertex. An efficient implementation of this for polygonal surfaces is available in VTK's vtkCellLocator class (Schroeder et al., 1997).

It is most convenient, for example to ensure that the face shapes are sampled with correct probability, to measure the components of $\mathbf{b}$ in units of standard deviations. The matrix $\mathbf{W}$ in steps 6 and 8 is thus a diagonal "unwhitening" matrix given by

$$\mathbf{W}_{ij} = \begin{cases} \sqrt{\lambda_i} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

In step 7, the method used for restricting the parameters depends on our model of the distribution. A reasonable assumption of a normal distribution tells us that we should limit $||\mathbf{b}||$ to be less than some k, where k^2 is given by the upper critical value of the χ^2 distribution with t degrees of freedom. For example, with our 400 examples, the model we computed requires $t = 40$ modes to represent 98% of the shape variation. If we use a significance level of $\alpha = 0.025$ (to include approximately 98% of the distribution) then we find $\chi^2(t, \alpha) = 59.34$ and so we should limit $||\mathbf{b}||$ to be less than 7.7 standard deviations. This method is encapsulated in the following rule:

$$\mathbf{b}' = \begin{cases} \frac{\sqrt{\chi^2(t,\alpha)}}{||\mathbf{b}||}\mathbf{b} & \text{if } ||\mathbf{b}|| > \sqrt{\chi^2(t,\alpha)} \\ \mathbf{b} & \text{otherwise} \end{cases} \tag{5.4}$$

For the threshold change between iterations, ϵ, we use a small multiple of a measure of the size of the template, defining:

$$\epsilon = 10^{-4} \times \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{v}_i - \overline{\mathbf{v}}|| \tag{5.5}$$

where $\mathbf{v}_i$ is a vertex in the template and $\overline{\mathbf{v}} = \frac{1}{n}\sum \mathbf{v}_i$ is the centroid of the vertices. Our starting template gives $\epsilon = 0.0051$mm.

When the fit does not correctly locate the target face it may be many iterations before the template change threshold ϵ is reached. To avoid this situation we additionally impose a limit of 1000 iterations on the fitting process. If this limit is reached we regard the fitting to have failed. Examples of this are shown later. The fitting process takes approximately one minute to run on standard desktop PCs.

As described in step 11, the LS-alignments of the template during the fit are initially carried out without scaling, using the Euclidean (rigid-body) transformation group. Since we built a size-and-shape model the template can vary in size to match the target but only to a limited degree (defined by the training set) and only where correlated with the shape of the target. Experiments showed that fitting with the similarity group can give greater accuracy in some cases but is less robust than using the Euclidean group (see Fig. 5.6). Thus after the template change drops below ϵ for the first time we extend the transform group, allowing the template to scale to better fit the target surface. When the change drops below ϵ for the second time we extend to the affine group, which again can give greater accuracy in some cases. The sequential introduction of larger transform groups works well in practice, sacrificing neither robustness nor accuracy.

In section 5.6 we will look at alternative schemes for controlling the amount of deformation and for exiting the loop. Firstly, we examine the robustness and accuracy of the fitting process as presented here.

5.5 Experiments

Before evaluating the performance of the fitting procedure, we illustrate it on an example face. Figure 5.3 shows the fitting in action on an image that is freely available on the web and that is not part of the training set. The target surface is shown in the first image with its original texture and then as a semi-transparent surface. The deformable template is shown with a texture in order to more clearly illustrate the convergence. Where the image looks mottled, the two surfaces are in close alignment. Figure 5.3 shows that the model is capable of converging to an unseen face scan which is not limited to the area modelled but extends beyond it. The fit takes under two minutes. (These preliminary results were presented in Hutton et al., 2001.)

Figure 5.3: The fitting in action on an unseen example, showing the target face scan, the initial placement of the template, an intermediate stage in the process and the final fit (side and front views).

To evaluate the robustness and accuracy of the fitting procedure described above, we tested it on the 21 unseen scans in the test set. The scans were manually landmarked in the same way as the ones in the training set, the 10 landmarks providing the ground-truth for measuring the accuracy of the fit.

There are several aspects of fitting performance that we need to test. Firstly, we look at how robust the fitting is to the position of the target as regards rotation and translation. We do this by altering the starting position of the template and measuring the accuracy of the final fit to a given target. Secondly, we look at robustness to shape variation by fitting to the different examples in the test set. To save time we run the starting position experiments on a single representative example, and the shape variation experiments from a single representative starting position.

5.5.1 Testing for robustness to position and orientation of the target

To examine the reliability of the fitting to the position of the target, an example was selected at random from the test set to be used as the target surface (#13 in Fig. 5.6(a)). We then ran the fitting procedure to termination many times, each from a different starting position. For the rotation robustness experiments, the mean template was placed in approximately the correct position by LS-aligning it with the known landmarks for the target surface, and was then rotated about one of the x, y or z-axes by a specified amount. The translation robustness experiments were carried out in a similar manner.

The locations of the ten landmarks for any template $\mathbf{x}$ can be computed using a

similar process to step 3 in Fig. 3.10 (p. 33). The landmarks on the base mesh are projected onto the template by finding the corresponding position in the same triangle in $\mathbf{x}$, using barycentric coordinates. Comparison of the predicted landmarks with the known target landmarks allows us to compute the RMS error, showing how successful the fit has been.

Rotation robustness

An illustration of the x, y and z rotation perturbations is shown in Fig. 5.4(a). A sample of the starting positions are shown translated along the x-axis for clarity, to avoid them all being on top of one another.

Figure 5.4(b) shows how the landmark accuracy after fitting is dependent on the degree of rotation of the initial position of the template. All three axes show clear basins of convergence - if the rotation perturbation is within $\pm 50°$ or thereabouts then the template converges to the same place, giving an RMS error on this example of 2.4mm.

Figure 5.4(c) shows the number of iterations before termination for rotations about the x-axis. Many of the cases beyond 50° reached the 1000 iteration limit and so would be regarded as failures. It can be seen that the starting positions that gave a low RMS error reached their final positions relatively quickly; within 400 iterations on this example.

Translation robustness

A similar experiment was run to test the reliability of the fitting to translation of the target. The mean template was again placed in approximately the correct position by LS-alignment of it with the known landmarks for the target surface, and then translated along one of the x, y or z-axes. Figure 5.5(a) illustrates the starting positions. The target surface is again shown in the centre. Note that it includes bits of surface other than the face region of interest. These are analogous to a cluttered background in an intensity image, in that they can confuse an algorithm that is searching for a particular pattern. The presence of the extra structure in the scene is reflected in the fitting results, where the accuracy for perturbations along one axis differs from one side to the other (Fig. 5.5(b)).

5.5.2 Testing fitting accuracy across a set of faces

The second aspect of the fitting that we tested was the robustness to shape variation. To do this we measured the RMS landmark error after fitting across the test set of 21 faces, from a single starting position given by $\overline{\mathbf{x}}$ in (3.2), ie. the mean position from the training set.

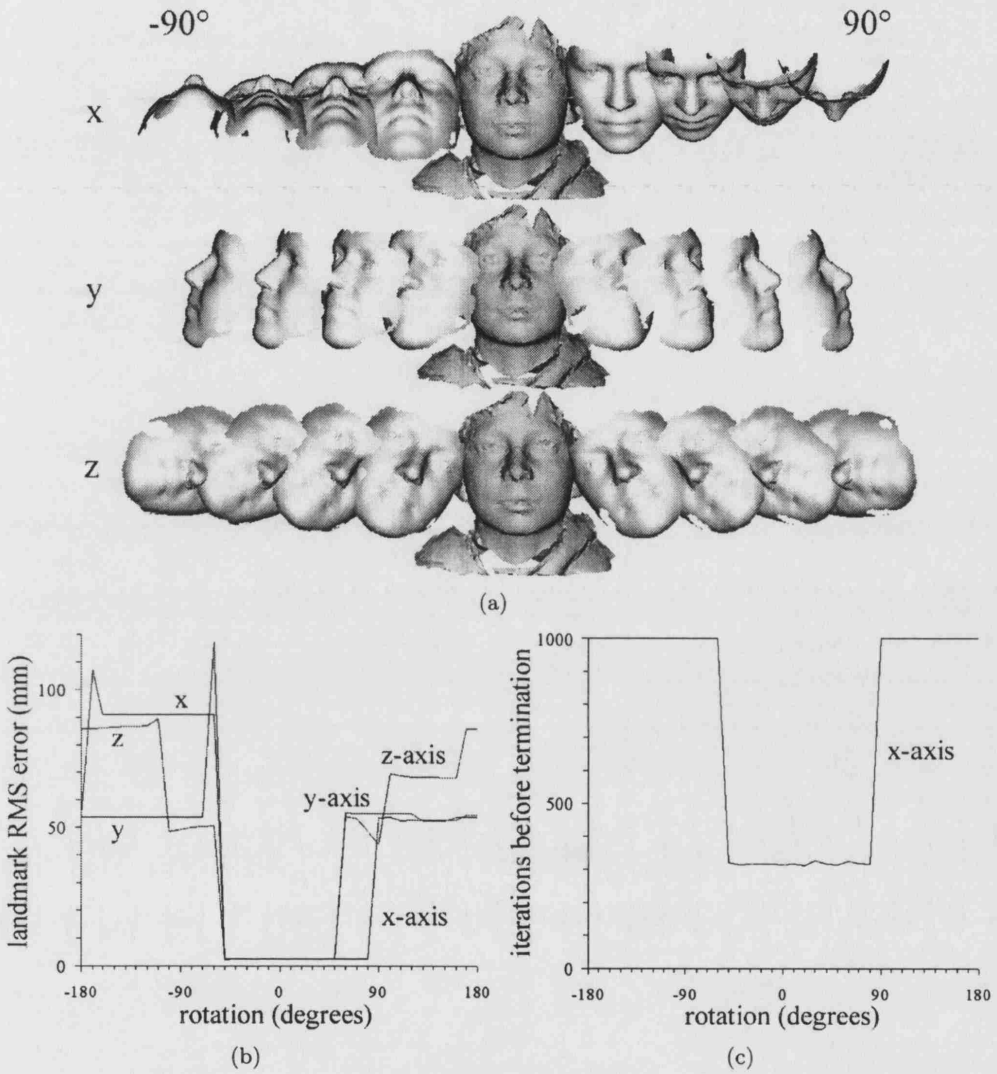
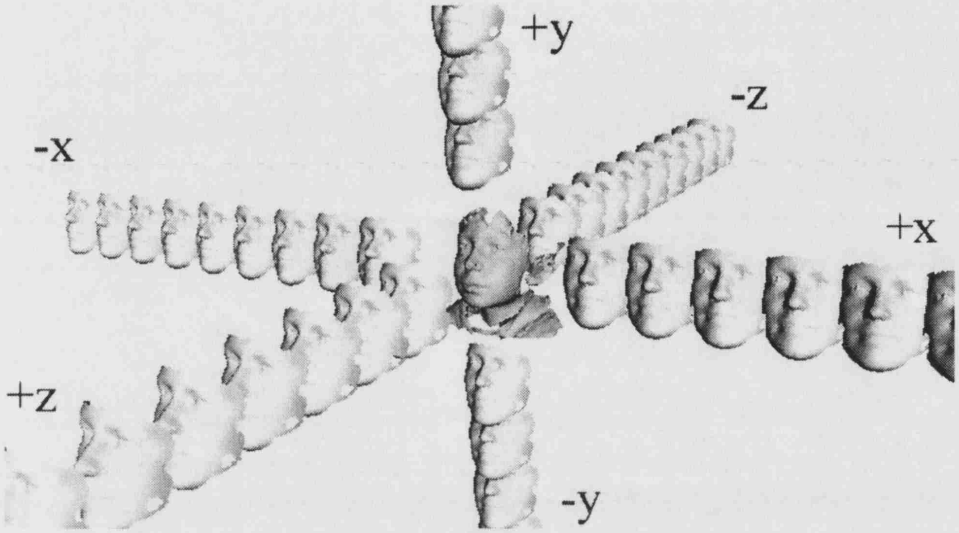
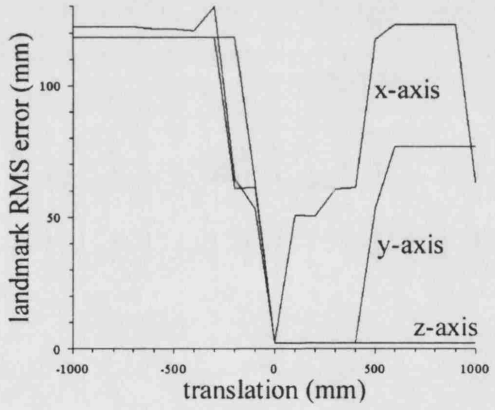


Figure 5.4: *(a)* Illustrations of how the template is rotated (x, y and z axes between $\pm 90°$) from the correct orientation before fitting commences, to test for rotational robustness. The templates are translated horizontally for clarity. *(b)* Graph showing the accuracy of the fit as a function of the rotation perturbation of the initial template (x, y and z axes). *(c)* The number of iterations taken before termination, for each of the starting positions after rotation about the x-axis. Rotations about the other axes gave similar results.



(a)



(b)

Figure 5.5: *(a)* Illustration of how the template is translated from the correct location before fitting commences, to test for translational robustness. The x-axis is side-to-side, the y-axis is up and down and the z-axis is front-to-back. The locations closest to the target surface are not shown for clarity. *(b)* The landmark RMS error after fitting from each of the starting positions.

Figure 5.6(a) shows the error after fitting for each of the 21 faces. The method of sequentially introducing larger transform groups as described in section 5.4 is method (C) in this figure. Fitting with this method gives an average RMS error over these 21 faces of 3.0mm, and a maximum of 5.5mm.

To demonstrate the effect of using other transform groups, we also ran this experiment using only the Euclidean group, method (A), and using only the similarity group, method (B). These methods did substantially worse, as can be seen in Fig. 5.6(a). Method (A) gave an average error of 5.0mm (15.8mm max), while method (B) gave an average of 5.8mm (59.2mm max). Method (B) gave an especially bad result on example 17. Disregarding this example, the average for method (B) was 3.1mm (5.2mm max).

To illustrate why method (C) is better, in Fig. 5.6(b) we look at how the template behaved during the fit for two examples (14 and 17). On example 14, method (A) does not get very close to the target, giving an RMS error of 11.2mm. Method (B) does much better, reducing the error very quickly to 3.96mm. Method (C) initially uses the Euclidean group and so follows method (A) to start with. After the introduction of scaling, method (C) reaches the same position as did method (B) but then with the introduction of the affine group the error drops further, finally reaching 3.2mm.

Example 17 shows a different situation. Here method (B) does not reduce the error but instead causes it to increase, the template is moving away from the correct position. Method (A) converges to the correct solution (the final error is 2.4mm). The subsequent introduction of larger transform groups has little effect on this example (but does not make the results worse).

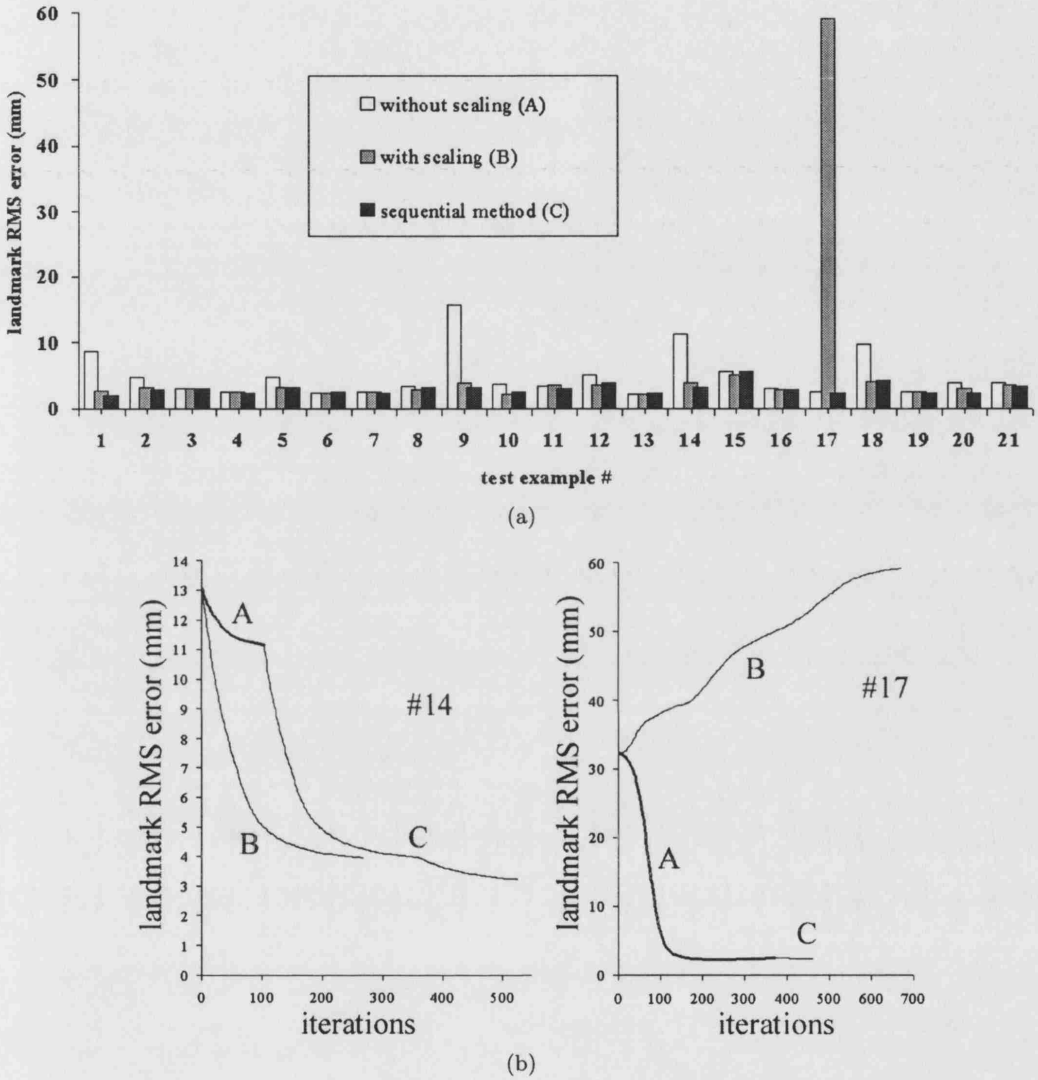


Figure 5.6: *(a)* RMS landmark error after fitting to the 21 faces in the test set. Three methods are compared: fitting with the Euclidean group only (A), fitting with the similarity group only (B) and fitting with the sequential introduction of Euclidean, similarity and affine groups (C). The sequential method (C) gives both robustness and accuracy. *(b)* The RMS landmark error over the fitting process for two examples in the test set: 14 and 17.

5.6 Alternative fitting schemes

There are several different ways of fitting a shape model to a target, and some subtlety in the interaction of the parameters that might be chosen. In Hutton et al. (2001) the method that we used was based on a fixed schedule of 100 iterations, with the number of modes varying from 0 up to the full complement of t modes (usually based on some percentage of the total variation, eg. 98%). This departure from the standard technique (using all t modes from the beginning, as in eg. Cootes et al., 1995) was found necessary because the parameters $\mathbf{b}$ were being limited in the wrong way. In one of the very earliest papers on ASMs (Cootes et al., 1992) can be found the following assertion:

> "Since the variance of b_i over the training set can be shown to be λ_i, suitable limits are likely to be of the order of $-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i}$ since most of the population lies within three standard deviations of the mean."

It seems to have become a not uncommon misconception that therefore the correct way to limit the variation to be legal is to limit each $|b_i| \leq 3\sqrt{\lambda_i}$. In most cases this is likely to give unsatisfactory results. Certainly it is true that, if the distribution is Gaussian, then *along any one axis* these limits are sensible but this does not take into account the multi-dimensionality of the model. Consider two modes, b_1 and b_2 that span a 2D Gaussian distribution of examples. By the rule just suggested, a synthesised example would be deemed perfectly valid if both modes had a value of 3 standard deviations (SDs). But then the total distance in parameter space of the synthesised example from the mean is $3\sqrt{2} = 4.24$SDs (Fig. 5.7). With only a few modes the effect of this is small but with 100 modes imposing a limit of 3SDs on each mode could permit examples that lie as much as 30SDs away from the mean! This effect becomes obvious if we try to fit a model using such a constraint on the mode parameters - the deformable template very quickly crumples up into a mess, a shape that is not a valid face.

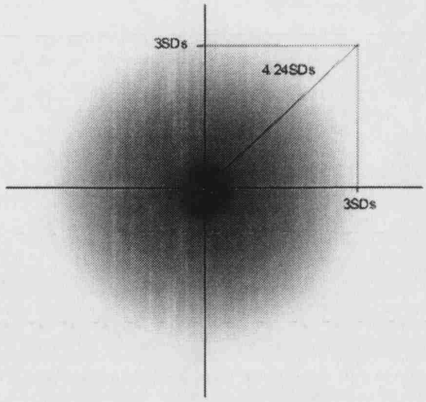


Figure 5.7: The multi-dimensionality of the model means that we cannot simply restrict each parameter to be within a certain range because the true distribution of the whitened shape parameters is assumed to be spherical.

If we assume that the distribution is a multivariate Gaussian then the correct method for constraining the parameters is as in (5.4) and is detailed in Cootes et al. (1994) and elsewhere.

5.7 Conclusion

In this chapter we have presented an algorithm for fitting a dense surface model to a new surface. We have examined the robustness of the fit to perturbation of the starting position and orientation and have looked at the accuracy of fitting over a set of faces. The method is not restricted to use on faces and while we haven't yet tested it ourselves on other data, we have received reports that the DSM fitting technique is being successfully used on laser-scans of the human ear-canal, as follow-on work from Paulsen et al. (2002).

Chapter 6

DSMs for Classification

This chapter looks at another application of DSMs, the use of the high-dimensional shape-space for finding the difference between different categories of faces, such as male-female.

6.1 Introduction

In the previous chapter we saw how DSMs can be used to register unseen face scans automatically. At the end of a successful fit, a dense correspondence is established between the new example and an existing population. This correspondence enables known properties of the population to be used to infer unknown properties of the new surface. For example, if the age of every person in the training set were known then the age of the new example could be estimated.

Medical applications of such a technique are numerous. Any condition that has a facial manifestation is amenable in theory to diagnostic assistance in this way. Candidates include:

- genetic syndromes that involve a facial dysmorphology, including: Noonan, Williams-Beuren, Velo-cardio-facial (VCF), Down, Rubinstein-Taybi, Sotos (and many others),
- non-genetic conditions that involve a facial component due to developmental processes, including: schizophrenia (although there may be a genetic component of schizophrenia, as in eg. VCF), prognathism caused by thumb-sucking, and
- conditions that affect the strength or the degree of control of the facial muscles, including: facial palsy (Ghattaura, 2001; O'Higgins et al., 2002).

Non-medical applications include:

- security (is the person in front of the scanner who they claim to be?),

- tracking (where exactly is the face of the person in front of the scanner?), and
- facial-expression analysis (what is the mood/lip positioning of the person in front of the scanner?).

Such applications have the potential to be more robust in three-dimensions than in two. For example, a front-portrait image-analysis security system could be fooled by a photograph whereas a three-dimensional image-based system would be much harder to fool.

The assumption behind the use of the shape-space for classifying faces is that faces with similar shape have a similar condition, ie. the shape-space is divided into (possibly non-contiguous, possibly overlapping) regions in which the properties of the faces are the same or similar: a male region, a smiling region, a Noonan Syndrome region etc. As discussed in Ch. 3, the metric or distance measure in the shape-space is the (partial) Procrustes distance (see eg. Dryden and Mardia, 1998), thus faces with similar shape are found close together.

There are many ways we might try to estimate unknown parameters for new faces. Classifiers for two-class or multi-class problems include: linear discriminant analysis, decision trees, neural networks, nearest-neighbour, closest-mean and support vector machines. All of these classifiers and more could be applied to face-shape data. In this chapter we merely illustrate the possibilities for classification and demonstrate the usefulness of the dense correspondence for improving the performance of the classifiers.

A technique that is proving popular for discrete classifications (where the problem is to classify unseen examples into one of two or more classes) is Support Vector Machine (SVM) classification (Vapnik, 1995). This has found recent favour for its treatment of classification as a structural risk minimisation problem, ie. the problem is expressed as finding the discriminant surface that maximises the separation of the classes. In this chapter, SVMs will be most frequently used. Another classifier that we sometimes use because of its simplicity is the closest-mean or minimum-distance classifier: each test case is simply given the label of the class whose mean in the shape-space is closest to it.

6.2 Two-class classification

To demonstrate the usefulness of DSMs for classification purposes, we first run some straightforward tests designed to show that the discrimination performance when using a dense surface is better than when using the hand-placed landmarks alone. This will show that the 'extra' data that DSMs use by establishing a dense correspondence has descriptive power.

6.2.1 Male-female classification

The dataset used previously (421 faces) was split into 10 cross-validation folds, keeping the numbers of male and female subjects in each fold approximately balanced. This was achieved by randomly sorting the the males and females separately and taking for each fold a successive 10th of each for the test set, the remaining examples becoming the training set. Each test set had 20 females and 21 males, and each training set 188 females and 192 males. We then built a dense surface model for the examples in each training set alone, thus ensuring that the test examples remained unseen. The use of cross-validation gives additional confidence that any performance figures reported are an accurate reflection of how the algorithm would perform when used on genuinely-unseen data. Cross-validation allows us to estimate the accuracy of our performance figures; the use of stratified (balanced) 10-fold cross-validation is a standard recommendation (Kohavi, 1995).

Then, in each fold, each training and test example was projected into the shape-space of the model using (3.9) (after first densely corresponding them with the base mesh). In each fold, an SVM was trained on the shape-space coordinates of the training examples (labelled as male or female) and then tested on the coordinates of the test examples. Comparison of the actual label with the label predicted by the classifier for each test example allows us to compute the overall accuracy of the classification. The figures for each fold are then averaged across all 10 folds to give an overall accuracy and standard deviation (SD). The results of this experiment are shown in Table 6.1.

fold	% of females classified correctly	% of males classified correctly	% overall accuracy
1	55.00	90.48	73.17
2	90.00	85.71	87.80
3	85.00	57.14	70.73
4	85.00	71.43	78.05
5	90.00	66.67	78.05
6	85.00	71.43	78.05
7	85.00	85.71	85.37
8	65.00	85.71	75.61
9	65.00	76.19	70.73
10	55.00	85.71	70.73
average	**76.0 (SD=14.3)**	**77.6 (SD=10.7)**	**76.83 (SD=6.0)**

Table 6.1: The accuracy of classifying faces into male or female using dense surface models.

In computing these results, the parameters for the support vector machines were varied. The radial basis function kernel was used each time (on other data it has been shown to give the best results (Kohavi, 1995)), with its width σ being given

by 5 common heuristics: Hinton, median separation, mean separation, Jaakkola and Jaakkola-mean (Table 6.2).

heuristic	σ given by
Hinton	square root of (data dimensionality / 2)
median separation	median separation of training data
mean separation	mean separation of training data
Jaakkola	median separation of each positive point to the nearest negative
Jaakkola-mean	mean separation of each positive point to the nearest negative

Table 6.2: Five different heuristics used to choose the RBF kernel width for use with support vector machines.

Additionally, the regularization parameter C was varied. The value of C determines the width of the soft-margin of the classifier, the penalty on any examples that are misclassified. Here we set C to take the values: 1, 10, 100, 1000. This gave 20 sets of results from which the best was selected. For the dense surface SVM, the mean separation heuristic with $C = 10$ gave the best results (the ones shown above), the full results are shown in Table 6.3. In our implementation, a weighting is routinely used to balance the number of examples from each class but this is not significant here since the numbers are almost equal. As an alternative to using heuristics for the kernel width, adaptive kernels (Burbidge, 2002) or even a simple grid search could be used, as recommended in Chang and Lin (2001).

The figures in Table 6.3 show that the performance of the classifier are not that sensitive to the parameters C and σ. In general, only a few percent separate the worst and best results found.

For comparison we carried out a similar experiment but trained the SVM on the raw coordinates of the 10 hand-placed landmarks, using the same training and test folds as before. For this dataset, the median separation heuristic with $C = 100$ gave the best results, these are shown in Table 6.4.

To summarise, DSMs plus SVMs gave a classification accuracy of 76.8% (std. dev. 6.0) while the 10 landmarks alone with SVMs gave 63.7% (std. dev. 7.0). This is a significant improvement in performance and is achieved through using the extra data present in surface scans of the face in areas away from the landmarks that can be placed by hand. This improvement is one of the key motivations for using dense surface models. Below we provide another demonstration of this for a different problem domain.

While gender classification is of little medical interest, it serves as an intuitive demonstration of the classification of 3D face scans. Recent work (Moghaddam and Yang, 2002) has also applied SVMs to gender-recognition using a large training set (793 males, 713 females on average) of cropped 2D frontal face photographs. They compared various classifiers and found that SVMs with a Gaussian RBF kernel (as

kernel width heuristic	C	males correct (%)	females correct (%)	accuracy (%)
Hinton	1	65.7	79.0	72.2
($\sigma = 4.58$)	10	73.8	78.5	76.1
	100	74.8	76.5	75.6
	1000	76.7	73.5	75.1
median	1	71.9	77.0	74.4
($\sigma = 2.52$)	10	76.7	72.5	74.6
	100	76.7	74.5	75.6
	1000	76.7	74.5	75.6
mean	1	74.3	75.0	74.6
($\sigma = 1.72$)	10	**77.6**	**76.0**	**76.8**
	100	75.7	75.0	75.4
	1000	75.7	75.0	75.4
Jaakkola	1	71.4	78.0	74.6
($\sigma = 2.59$)	10	77.1	75.0	76.1
	100	77.1	74.5	75.9
	1000	77.1	74.5	75.9
Jaakkola-mean	1	72.4	75.5	73.9
($\sigma = 1.78$)	10	77.1	74.5	75.9
	100	76.2	74.0	75.1
	1000	76.2	74.0	75.1

Table 6.3: The accuracy of classifying faces into male or female using dense surface models, with five different heuristics for σ and four different values of the regularization parameter C. The values of σ shown in brackets are from one fold, and are included for illustration only since they vary between folds.

fold	% of females classified correctly	% of males classified correctly	% overall accuracy
1	75.00	42.86	58.54
2	60.00	42.86	51.22
3	65.00	66.67	65.85
4	65.00	71.43	68.29
5	55.00	61.90	58.54
6	45.00	71.43	58.54
7	80.00	71.43	75.61
8	70.00	57.14	63.41
9	70.00	66.67	68.29
10	60.00	76.19	68.29
average	**64.5 (SD=10.1)**	**62.9 (SD=11.8)**	**63.7 (SD=7.0)**

Table 6.4: The accuracy of classifying faces into male or female using the 10 hand-placed landmarks only.

used above) performed best, achieving a classification rate of 98% for males and 95% for females. The accuracy that they achieved is far greater than that reported in the experiment above. Other than the fact that Moghaddam and Yang used a much larger data set, one possible reason for this difference is that there is more gender information in a grey-level photo than in the shape of the face as captured in a surface scan. In the next chapter we look at how grey-level information might be incorporated into our 3D face model. Another advantage of Moghaddam and Yang's dataset is that all their subjects are adult. We will show later in this chapter that the features that discriminate gender become more pronounced with age. Additionally, the scans in our dataset were acquired on the DSP400 scanner, a relatively low-resolution scanner. A set of better quality scans would likely improve the performance.

6.2.2 Noonan Syndrome classification

As a more convincing demonstration of how the extra information not captured by the landmarks on a surface is useful for classification, we ran a similar set of experiments to those above to distinguish children and adults with *Noonan Syndrome* (Allanson et al., 1985) from controls[1]. Noonan Syndrome (NS) affects the shape of the face (it is a *facial dysmorphic* syndrome) as does the more common Down Syndrome. An individual with NS is shown in Fig. 6.1 (a boy whose family has given explicit permission for the image to be used in research literature). NS has an associated heart condition that requires treatment but it is often the facial morphology that suggests a diagnosis. The nature of NS and similar syndromes makes them a good test of dense surface models, as well as providing a specific medical application for the work. While there is now a genetic test for some instances of NS (Tartaglia et al., 2001), this test is not completely effective. Otherwise, the diagnosis is made by an experienced clinical geneticistassessing multiple aspects of the patient.

Dense surface models were built from a set of 573 controls and 101 NS subjects, with 11 hand-placed landmarks on each. 10-fold cross-validation was used with an SVM classifier (the best settings found through experiment were: Gaussian RBF, the Hinton heuristic, $C = 1$). To produce an ROC curve (Swets, 1988), the decision values for each test example were ranked in size order. Using the label of each test case (either positive or negative) allows the ROC to be directly plotted.

Each point on the ROC is averaged over the 10 folds, by sampling along the false positive axis (Fawcett, 2003). From each model, enough modes to capture 99% of the variation were retained, typically 80. In the previous experiments, 98% was used but later experiments (see Fig. 6.4) showed that more modes typically improve the classification performance. The results are shown in Fig. 6.2.

[1]This work was reported in Hammond et al. (2004b) and elsewhere.

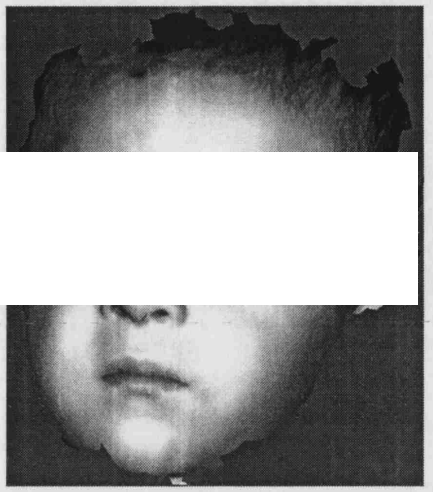


Figure 6.1: A boy with Noonan Syndrome. While the differences associated with NS are not obvious, they include: a downward slope to the eyes, the eyes are further apart, the bridge of the nose is depressed, the mouth is small, the lower face is underdeveloped and triangular.

In summary, representative results obtained when using 11 landmarks only are: sensitivity=83%, specificity=83%; while for the surfaces: sensitivity=91%, specificity=91% can be obtained.

The comparison between different shape descriptions, above, can only be made quantitative in the context of a full application of the technique - in this case classifying a facial dysmorphic syndrome. Without such an application as the end goal, there are no absolute standards by which to compare two models, hence the fact that in earlier chapters we were unable to justify the number of landmarks used. Instead of asking: "what is the optimum number of landmarks?", a question that is not answerable, we can now ask: "what is the optimum number of landmarks for discriminating the faces of individuals with Noonan Syndrome faces from those of controls?", a question that is in principle answerable by comparing the performance of different models. Figure 6.3 compares three such models, built with 11, 19 and 57 landmarks. There is no clear difference between these three ROCs, although the model built with 11 landmarks performs slightly worse than the other two. The choice of which model to use depends on the relative cost of either not identifying an individual with Noonan Syndrome or of falsely identifying a control as having Noonan Syndrome and having to run further tests. In a clinical setting such a decision, in principle, comes down to a balance of costs and quality of life assessments.

If a balance between sensitivity and specificity is desired then the model built with 19 landmarks performs best. It is not clear why the model built with more landmarks (57) performs less well here. This may be due to errors in placing the additional

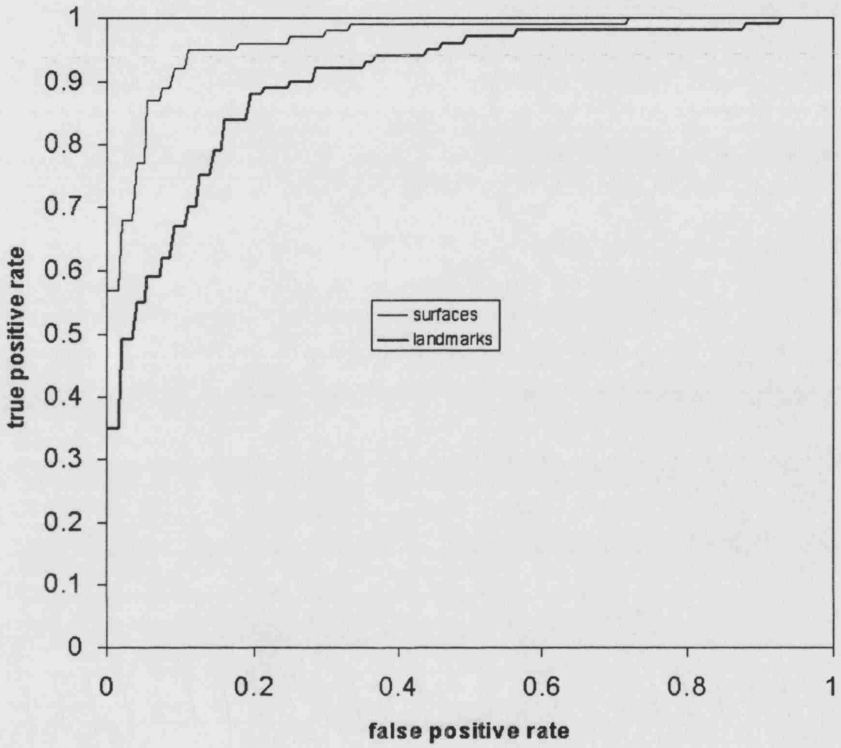


Figure 6.2: ROC curves comparing the performance of classifying Noonan Sydrome faces from controls. The top line shows the performance of a DSM trained using 10-fold cross-validation on a set of 573 controls and 101 subjects with Noonan Syndrome. The examples were classified by a support vector machine (Gaussian RBF, the Hinton heuristic, $C = 1$). The bottom line shows the same result if the 3D coordinates of the 11 landmarks are classified using SVMs in the same way, after Procrustes alignment to remove rotation and translation differences. This ROC clearly demonstrates how DSMs allow the 'extra' shape information present between the landmarks on a surface to be incorporated into a model.

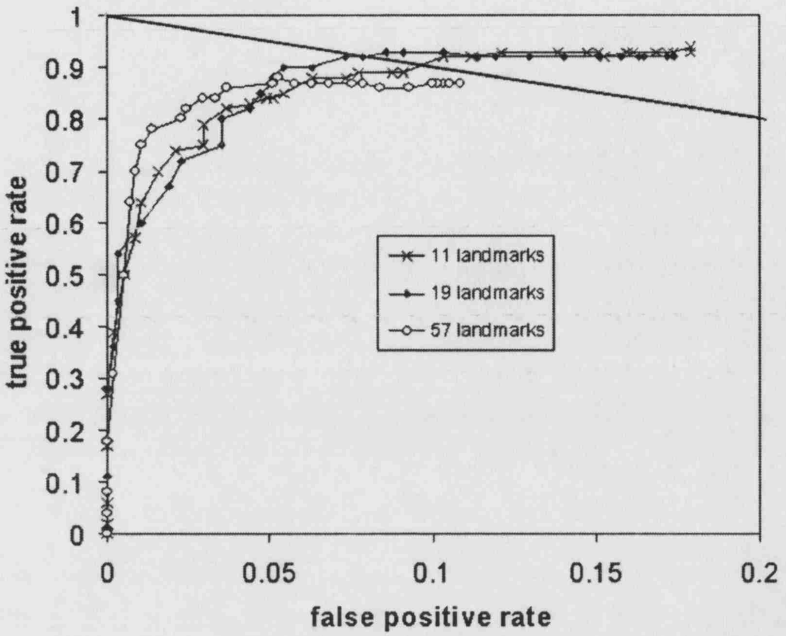


Figure 6.3: ROC curves comparing the performance of classifying Noonan Syndrome subjects from controls in three models, built with 11, 19 and 57 landmarks on each face. The straight line shows where the sensitivity and specificity are balanced. Note that the axes are scaled differently.

landmarks, since they cannot be as precisely localised as the original landmarks. It is also possible that if too many landmarks are used, the Thin-Plate Spline interpolation may create distortions in the surface that cause the model to perform less well, although such effects have not been seen in synthesised surfaces. The best performance obtained with the 19 landmark model is a sensitivity of 92% and a specificity of 93%. The lack of an obvious difference between the 19 and 57 landmark models is a good sign that a small set of landmarks is sufficient to achieve a good correspondence between the surfaces. In general, a smaller set of landmarks is preferable to minimise the time required to landmark the scans and also to compute the TPS warping of the surfaces.

However, this experiment does not justify the choice of *which* landmarks to use. With sufficient computational resources, different combinations of landmarks could be tested for optimality with regard to classification. A similar procedure is used in Davies et al. (2002a) to establish the dense correspondence between surfaces (by treating all the vertices as landmarks) with regard to an overall description length criterion. Such methods may well improve the classification performance of DSMs at the cost of additional computational time to build the model.

6.2.3 Varying the number of modes

In some of the above experiments we used 98% of the variation (approximately 40 modes for a model built with 421 faces). This is a typical rule of thumb used to exclude noise from the model, since the remaining 380-odd modes account for only a tiny amount of the total variation. Now that we are considering a direct application of our PCA model, namely for classification, we can examine how useful the modes are by considering how the classification performance varies as a function of the number of modes we retain. Figure 6.4 shows the result of an experiment where an SVM and a closest-mean classifier were trained for male-female as before, but on different numbers of modes.

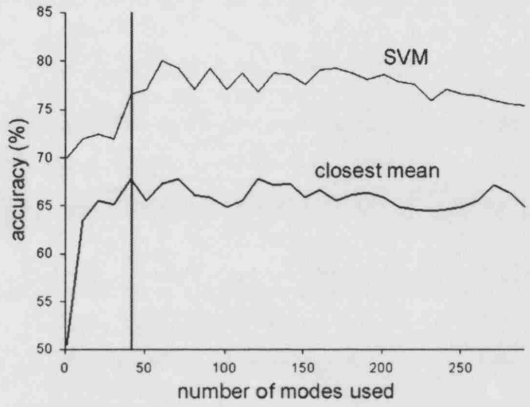


Figure 6.4: The overall accuracy of male-female classification accuracies using dense surface models and a varying number of modes from 1 up to 300. The results of two classifiers are shown, SVMs and the simple closest-mean classifier. The number of modes required to model 98% of the variation is 42, indicated by the vertical line.

For the SVM an underlying trend can be seen - the performance rises to a peak and then slowly descreases. The peak is somewhere between 50 and 100 modes, and the 98% cutoff of 42 modes seems to be suboptimal, representing an overly-compact model. For the closest-mean classifier, the 98% cutoff seems reasonable since the performance does not improve beyond this point. However, the closest-mean classifier performs less well than the SVM on this problem.

An important observation is that cross-validation could be used to select the number of modes for a given classification task, in a similar manner to the way σ was selected (section 6.2.1). For example, to classify gender using SVMs we might use Fig. 6.4 to decide that 60 modes should be retained, since the remainder are not useful for the task in hand and in fact degrade the performance. More modes means more storage space required, and slower execution times - if these considerations were important in the application (for example, online classification) then fewer modes might be preferred.

6.2.4 Error rates versus age

It is a common observation that some faces are more easily identified as male or female than others. It seems likely that this ability to discriminate is age-dependent, since many of the distinguishing features of gender become more pronounced in particular with the onset of puberty. To see if we can detect this effect in our classification using DSMs and SVMs, we can plot the classification rates from the experiment above against the age of the individual (Fig. 6.5). It can be seen that indeed the classification performance does increase with age, with young faces being particularly difficult to correctly classify.

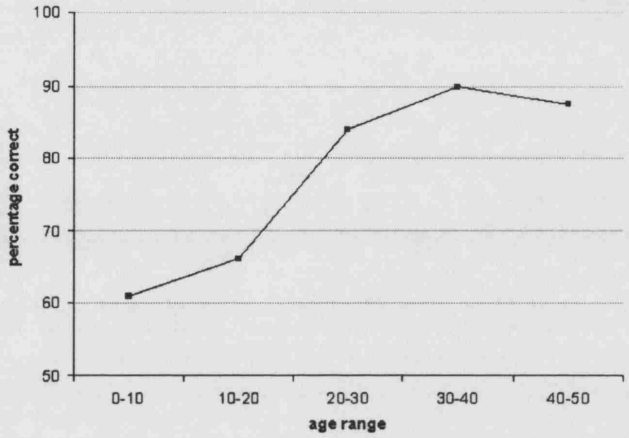


Figure 6.5: The classification rates for gender vary dramatically over the age of the individual, with older people in general being easier to classify as either male or female.

6.3 Modelling age

Having seen that age is a confounding factor in modelling human faces, we need to investigate further how it might be understood and modelled. If the age of each individual is known then it could well be useful to incorporate it into the classifier, allowing the classifier to make a better estimate of (for example) the person's gender knowing their age. Medically, the effects of age are important since many genetic conditions (eg. Noonan syndrome) manifest effects that change with age, as the face develops. If we are fully to understand the differences caused by genetic conditions such as these then we need not only a model of normal aging but also a model of aging in Noonan Syndrome. In this section we describe one approach that can be used to model aging.

The face changes throughout life. In young adults there is considerable growth of the skeletal structures, mostly in the lower face, along with an increase in muscle tissue and changes in the volume of fatty tissues. In middle life there is little change in the bone structure but continued growth in cartilage, especially in men, affecting amongst

other things the shape of the nose. In later life, changes in both muscle tone and skin elasticity affect the outer shape of the face considerably. To model these complex changes we will need a large amount of data on which to train a model. Ideally we would use longitudinal data, ie. 3D scans of the same people at different times throughout their lives. Then for each we could plot the path through shape-space associated with the aging process - their *aging trajectory*. However, since the technology for 3D scanning has only been widely available for the last few years we do not have any longitudinal data. Also, even if such data *were* available, a large number of subjects would still be necessary in order to obtain a robust estimate of overall growth patterns despite the individual differences.

Longitudinal studies of face aging have been carried out using retrospective photographs (Lanitis et al., 2002). With 10 or so time samples for each individual, their trajectory through the parameter space of an appearance model was modelled as a linear, quadratic or cubic curve. They suggested that aging could be simulated for a new example by using a weighted combination of the aging trajectories for the subjects in the dataset, based on appearance or lifestyle similarities. They demonstrated that the age of an individual could be used to improve the performance of a face recognition system.

Additionally, many longitudinal studies of landmark position have been undertaken, using morphometric techniques. These include work on face profiles from laser scans (Morris et al., 1999a,b), primate skulls digitised with a robot arm (O'Higgins and Jones, 1998; O'Higgins et al., 2002), human skulls from specially calibrated side and front x-rays (Dean et al., 2000) and other organs.

One study has used a dense correspondence to analyse growth of the human mandible (Andresen et al., 2000). CT scans of six subjects (4 male, 2 female) up to the age of 12 were used. Normally CT scans cannot be taken of healthy subjects because of the radiation dosage but these were children born with Apert's syndrome, a genetic condition that affects the growth of the cranium but not the mandible. This study found that the growth in all the subjects was consistent with a linear model.

A significant problem with almost all such studies to date has been the sparsity of available longitudinal data. If enough data were available, then after registration the bundle of trajectories through the parameter space might look something like Fig. 6.6. Growth analysis would then be a process of finding models for aging that fitted the data well, by incorporating the confounding factors such as gender (as mentioned), lifestyle, ethnic group, diet and genetic conditions.

Methods for inferring underlying trends in data are in general known as *regression*. In Lanitis et al. (2002), various types of polynomial regression were used, from linear to cubic. However, modelling more complex curves in this way requires high numbers

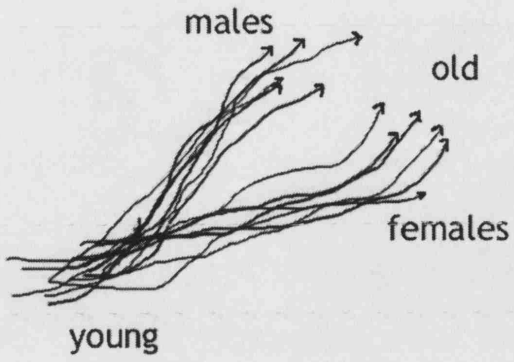


Figure 6.6: A cartoon depiction of how the trajectories for a population might look, if such data were available. While there would be underlying trends in the movement, both globally (young to old) and by subgroup (males v. females), the trajectory for each individual would differ from those of other individuals.

of data points in order to obtain robust estimates. A non-linear regression method with the potential advantage that it can cope with more complex curves is *kernel smoothing*. Here, a sliding average is taken for a range of ages, using a weighting kernel. A sophisticated technique more recently developed is *support vector regression* (SVR) (Drucker et al., 1997), which extends the support vector paradigm from the classification problem to the regression problem.

With kernel smoothing, different kernels can be used but in general give trajectories that are very similar. Kernels with limited extent, such as the box kernel or the triangular kernel, can give rise to sharp jumps in the trajectory, unlike kernels with infinite extent such as the Gaussian kernel, but are more efficient to compute. The application of kernel smoothing to face shape data was presented in Hutton et al. (2003b). Figure 6.7 shows the trajectories computed for a set of 200 males ranging in age between 0 and 50, using triangular kernels of different widths.

The path of the average aging trajectory is parameterised by the target age, t, and given by:

$$\mathbf{a}(t) = \frac{\sum_{i=1}^{n} w(\text{age}_i, t)\mathbf{b}_i}{\sum_{i=1}^{n} w(\text{age}_i, t)} \tag{6.1}$$

where n is the size of the population, and age_i and $\mathbf{b}_i$ are respectively the age and location in shape-space of the i'th subject.

The triangular kernel is defined by:

$$w(x, t) = \max(1 - \frac{|x - t|}{\text{width}}, 0) \tag{6.2}$$

where in Fig. 6.7, width $\in \{5, 10, 15, 20\}$. To avoid the effects of insufficient data (the meandering of the trajectory at the older end), we selected the width of 20 years for

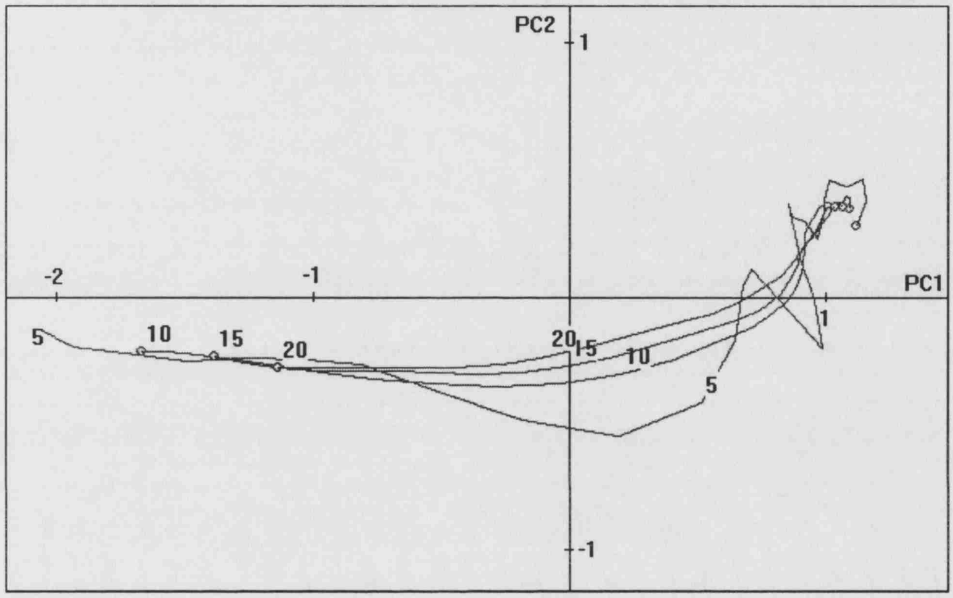


Figure 6.7: The trajectories through shape-space, averaged from a set of 200 male subjects. Different kernel widths between 5 and 20 years give progressively smoother trajectories. The trajectories are plotted against the first two principal components (see Fig. 6.10). Wider kernels cause the ends to be drawn in, giving shorter trajectories - we account for this effect by correcting the age at each point (see text). To avoid the effects of insufficient data we choose a large kernel width, 20 years, for the rest of the work in this section.

the following experiments.

One effect of using kernel smoothing is that the width of the kernel limits how close to each end of the parameter range estimates can be made. For example, when we compute $\mathbf{a}(0)$ (new-born baby) we will get a weighted average that will look too old since only older faces were available in the average. This results in a face that is older than one would expect at the lower end, and a younger than expected face at the upper end of the age range. We can compute a value for a more realistic age of each average using:

$$\text{age}(\mathbf{a}(t)) = \frac{\sum_{i=1}^{n} w(\text{age}_i, t)\text{age}_i}{\sum_{i=1}^{n} w(\text{age}_i, t)} \tag{6.3}$$

The plot of $\text{age}(\mathbf{a}(t))$ versus t in Fig. 6.8 shows the typical rounding-off at each end of the age-range. Since $\text{age}(\mathbf{a}(t))$ is monotonically increasing with t we can use this relationship to infer a unique t value for a desired age. For example, to construct an acceptable 10-year old average face we need $t = 7.3$ for the male subgroup and $t = 3.4$ for the female subgroup. This difference is caused by the varying numbers of examples at different ages for the two sexes. The overall age distribution is shown in Fig. 6.9.

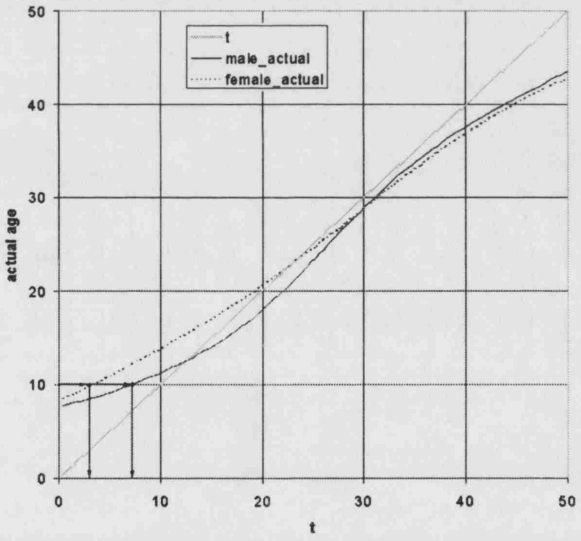


Figure 6.8: A plot of $\text{age}(\mathbf{a}(t))$ versus t shows how the extremes of the age-range are rounded-off. At the lower end the faces are older than their corresponding t values while at the upper end they are younger. We allow for this by using the value of t needed to obtain the face of the correct age.

By using this relationship, we can annotate the trajectories with their corresponding ages (Fig. 6.10). This plot shows some clear differences between the genders, with males growing to a bigger size (mode 1) than females, and in general being further towards

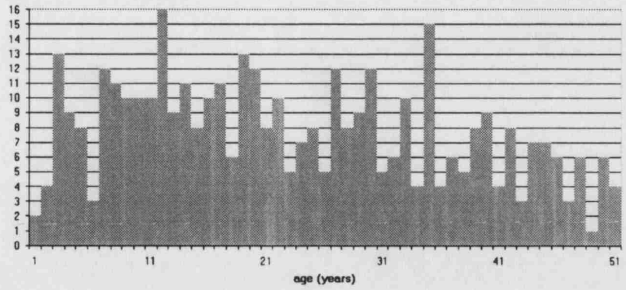


Figure 6.9: The age distribution for our dataset of 400 subjects, both males and females.

the negative end of mode 2 in early life.

Faces from along these trajectories are shown in 6.11. While there is not a dramatic visible difference between them, several features can be noted, included a size difference in males, most noticeably above the age of 20, and a continual nose elongation in the male faces that is not present in the female faces. Shown as an animation, the male-female differences are more striking.

One obvious question that is thrown up by these results is whether the differences found are significant. It is clear from Fig. 6.10 that the overall distribution, at least on these two axes, overlaps to a large extent. In fact the same is seen if we plot different pairs of axes, for example the first mode against the third (see Fig. 6.12). A standard method for estimating the reliability of an inferred value such as a mean is *bootstrapping*. A set of bootstrap samples is found by sampling *with replacement* from the dataset. This means that, for N subjects, a sample is taken by choosing N subjects from the set, by choosing at random each time. The probability of duplicates in the bootstrap sample causes the inferred value (in this case the mean) to vary, and this distribution tells us how significant the difference between the trajectories is. For the theory behind why this works, see eg. Davison and Hinkley (1997).

We computed the trajectory points for 10,000 bootstrap samples. From the distribution of the bootstrap samples around each point on the trajectory (not shown) we can infer a Gaussian and estimate its mean and standard deviation. Plotting ellipses at 2.45SDs gives us the area in which 95% of the distribution lies[2]. These ellipses are shown in Fig. 6.12. Where the trajectory is more uncertain, the ellipses are bigger. From this graph it is clear that the trajectories *are* significantly different, especially towards the adult end of the age-range, since the means of each gender lie outside of the uncertainty ellipse of the other gender at the corresponding age.

The separation in shape-space between the genders clearly increases with age, as

[2]We have to use 2.45SDs for these ellipses since the distribution is projected into a two-dimensional space. For a one dimensional Gaussian the more familiar 1.98SD limit gives us 95%, but in n-dimensions the χ^2 table must be consulted.

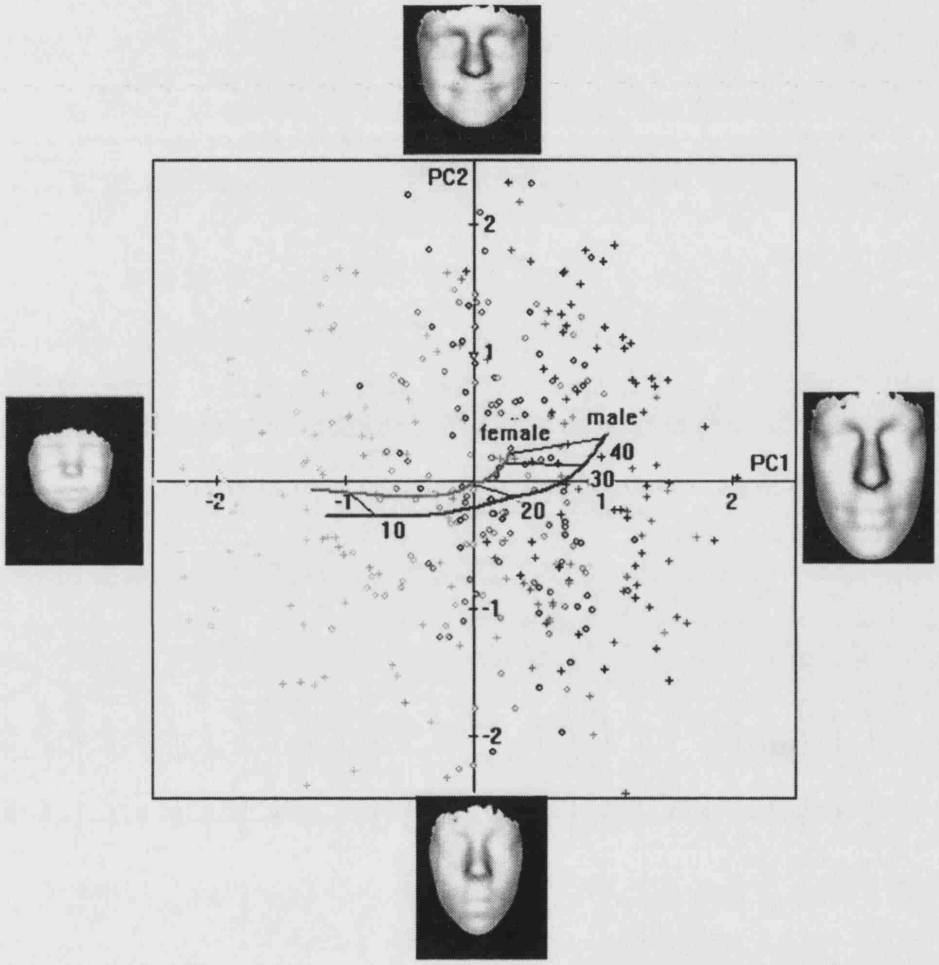


Figure 6.10: The average growth trajectories for male and female, with labels marking the ages in years. The axes are the first two principal components, in standard deviations. The ±3SD eigenfaces are shown at the end of each axis. The scatter plot in the background shows the overall distribution of males (crosses) and females (circles).

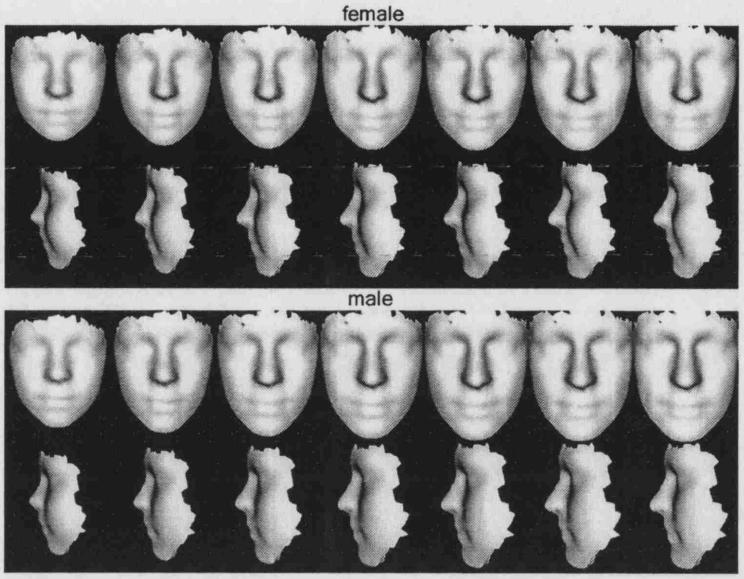


Figure 6.11: Synthesised faces at ages 10, 15, 20, 25, 30, 35, 40 along the aging trajectories for females and males. The scale is the same throughout.

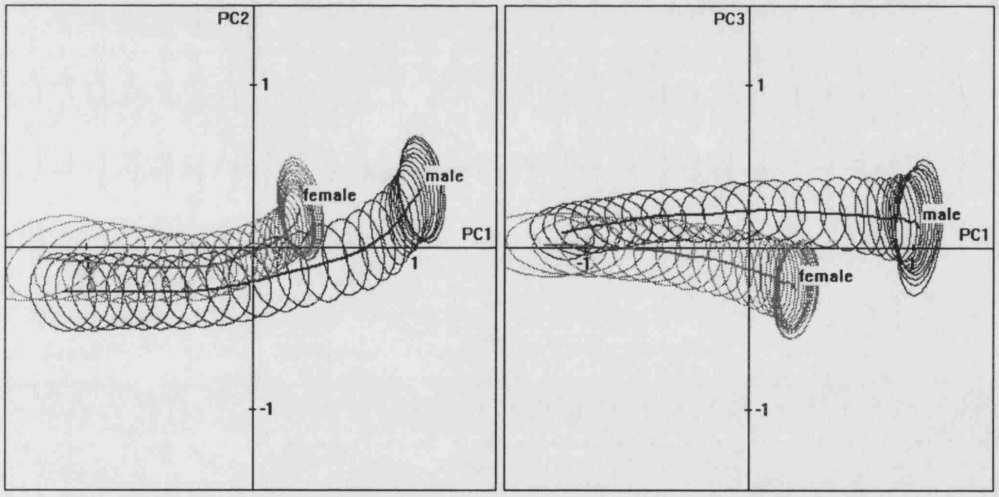


Figure 6.12: The average growth trajectories for male and female, with 95% confidence regions on the trajectory points represented as ellipses. The plot on the left shows the trajectories against the first two principal components, the one on the right shows the first against the third. The axes are in standard deviations.

can be seen for the first two principal components in Fig. 6.12. To see the effect using all 40 dimensions, we computed the (Euclidean) distance between the two, again using bootstrap samples to estimate the 95% confidence margin. The results are shown in Fig. 6.13. Between the ages of 7 (the lowest we can reliably model with this dataset) and about 12 the trajectories are reasonably parallel, but after this they move apart steadily. It is interesting to note that the trajectories continue to move apart throughout life, not just during the pubertal growth spurt, confirming that the soft tissues in males and females age differently.

The fact that the 95% confidence margins do not overlap the zero distance seems to suggest that the trajectories are indeed well separated. However, it is not clear if taking the difference between the trajectories for different bootstrap samples gives an accurate estimate of the actual error, since the bootstrap samples may not be linearly distributed. In the published paper (Hutton et al., 2003b) the confidence regions were not included because of a suspicion that these results might be misleading[3].

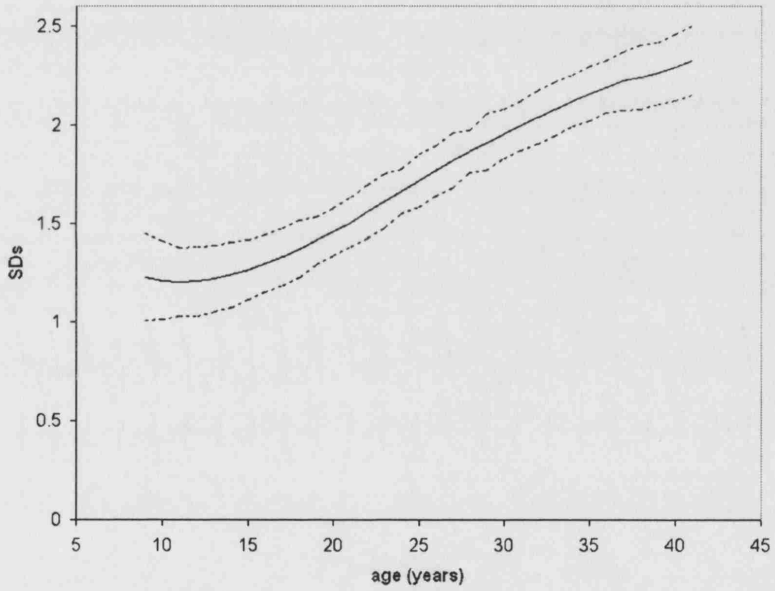


Figure 6.13: The separation between the two gender trajectories, computed in the full 40-dimensional shape-space. The dotted lines show the 95% confidence margin, computed from the bootstrap samples. From the age of 12 or so the trajectories move further apart.

[3]Henry Potts, personal communication.

6.3.1 Age estimation

One method for validating our model of aging is to use it to estimate the age of unseen examples. A way of doing this is to find the closest point on the overall aging trajectory for each test case, and assign to it the age represented at that point. This assumes, however, that the distribution of examples around the trajectory points is the same in every direction, and it can be seen (since the distribution of each boostrap mean reflects the distribution of the examples) from Fig. 6.12 that this is not the case. A superior method, therefore, would be to find the smallest Mahalanobis distance (using the covariance matrices that gave the ellipses in Fig. 6.12) from each point on the trajectory and use that age instead. The plot of Mahalanobis distances against age for a test example would indicate the confidence in the age estimate - a sharp peak would imply high confidence.

This could be done for the overall trajectory (both genders together), or for the genders separately. With enough data the gender-specific model should work better but for a small dataset, the overall trajectory might be better defined and thus give better results.

Alternatively, standard regression techniques could be used to estimate the scalar parameter (age, in this case) from the test example coordinates. The LIBSVM package (Chang and Lin, 2001) implements support vector regression, and it is straightforward to obtain age estimates for the test examples, in the same way that we obtained class-estimates for the male-female classification problem dealt with earlier. Figure 6.14 shows the actual and predicted ages of 41 test cases, predicted from a DSM model built with 380 training examples. The mean absolute error was 7.07 years; the RMS error was 10.4 years.

This regression is carried out without using the gender of the examples in any way. We have seen in Fig. 6.10 that the growth patterns are different for the two genders, and thus a method of incorporating gender into the age-estimator is likely to improve its performance. One simple way is to perform *gender normalization* as shown in Fig. 6.15. In this normalisation procedure, the difference vector between each example and its class mean is added to the overall mean to compute a gender-normalized location in shape-space.

If we gender-normalize the examples in our training and test sets before applying SVR, we get a slight improvement in performance; the mean absolute error becomes 6.99 years, while the RMS error becomes 10.2 years. These results are not significant.

Further experiments would determine whether any of the other methods discussed could out-perform these figures, but age estimation is not a central goal of our work and so we leave these ideas, possibly for future investigation, and move on.

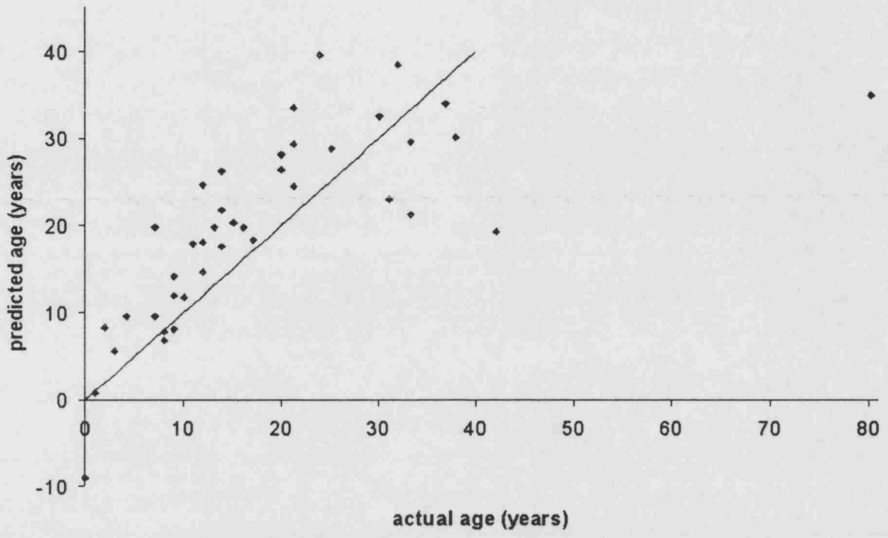


Figure 6.14: Age estimation results using ν-support vector regression (Schölkopf et al., 2000) with a linear kernel (ν=0.5, C=1) on a training set of 380 examples, ignoring gender. The mean absolute error is 7.07 years.

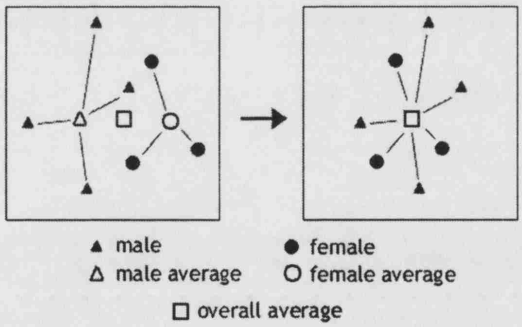


Figure 6.15: A simple method for class-normalization, shown with gender as an example. The vector between each example and its class mean is added onto the overall mean to compute a class-normalized location in shape-space.

6.3.2 Age morphing

The proposal made above was that understanding age could improve the performance of our classifier. To see if this is the case we need to normalize each face for age, to *remove* the effects of age on a face, leaving us with faces that we can directly compare. To do this we need to know not just how the average face ages but also how any given face will age. While in general we know that the aging trajectories will be different for each person (Fig. 6.6), an obvious assumption to make is that for a particular subgroup they are parallel. This assumption (also considered in Lanitis et al., 2002) is made in the absence of longitudinal data to the contrary and could be relaxed if longitudinal data becomes available.

An illustration of how we can use a parallel trajectory to change the age of a face is shown in Fig. 6.16. Since we assume that the aging trajectories are parallel we can simply add the relevant portion of the average trajectory for the selected subgroup (males in this case) to the location of the subject's scan in shape space.

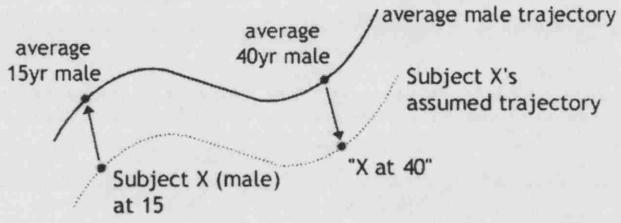


Figure 6.16: How to change the age of a face if we assume that aging is parallel to the average. The average trajectory for the selected subgroup (in this case: males) is added onto the position of a given face in shape-space to compute a new face.

We used this approach to show the effects of growth on the face for a BBC documentary first screened in 2002 called 'Teen Species'. Figure 6.17 shows some of the images that were used in the programme - a scan of a teenager (centre) is age-morphed backwards and forwards in time[4].

6.3.3 Age normalization

Using the age morphing method described above, we can obtain an age-normalized equivalent for a face, by morphing along the trajectory to a specific age point, eg. 25 years. For the purposes of improving the classification, however, we do not need to do this, since we are only interested in the differences between each example and the 'expected' face, given their age. This difference is represented in Fig. 6.16. By subtracting from the location of each example in shape-space the average face location

[4]Some animations from the programme can be found at the BBC's website: `http://www.bbc.co.uk/science/humanbody/body/interactives/lifecycle/teenagers/` (click on the face)

Figure 6.17: An example of age manipulation. Here the face of a teenager is warped back to the age of a toddler, and forward to that of an adult. The grey-level texture in these images is faked by artistically blending with images of individuals - it is the shape-change that is of interest here. With a combined model of grey-level appearance and shape (see the next chapter) it would be possible to do this properly, although facial appearance is the net result of many factors, only some of which are predictable.

at the relevant age we can project each example into an age-normalized space. For example, from the location in shape-space corresponding to a subject at 15 we would subtract the average 15 year old face location. In this space we might expect to obtain better classification for age-correlated factors such as gender. Furthermore, if sufficient data were available, by age-normalising using the trajectory for the correct gender we would expect to obtain improved classification accuracy for factors that are correlated with both age and gender.

To test this possibility, we compared the classification rates for *Noonan Syndrome* when using the above age normalisation procedure and when not using it. Using a triangular kernel of width 20 years (as throughout), we ran 10-fold cross-validation using an SVM with the Hinton heuristic and C=1. To produce ROC curves, the relative misclassification cost was varied as before. The results of this experiment are shown in Fig. 6.18.

These results are somewhat surprising, since it appears that age normalisation is of no benefit in this case. It appears that either the SVM classifier is already able to separate the effects of age from the effects of the syndrome, or that our age trajectories are over-smoothed and more data is required.

Age normalisation *may* be of benefit for some datasets however, such as when few

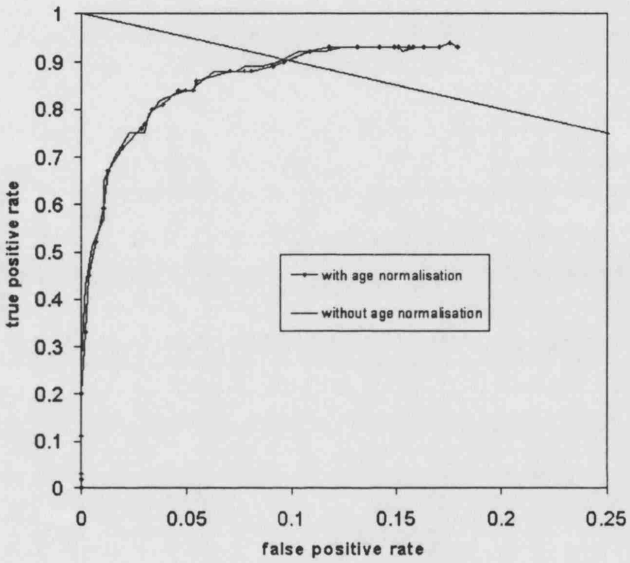


Figure 6.18: ROC curves for classifying subjects with Noonan Syndrome from controls, comparing the classification performance obtained with and without the use of age normalisation. The two curves are essentially identical, leading to the conclusion that age normalisation does not improve classification of this condition on this dataset. The ROCs are not smooth since only 10 samples at each point are taken. More splits of the data would give smoother curves at a greater computational cost. The scales of each axis are again different.

examples of a specific age are available for training. This could be tested through the creation of a sparse dataset, with deliberate gaps in the age distribution. However, once again such considerations are secondary and thus these ideas are not explored further here.

It is interesting, nonetheless, to look at the effect of the width of the kernel used to produce the growth trajectory on the classification rate. In Fig. 6.7 we visually compared the effect of using different widths but were unable to give an objective justification for the use of any particular width. Figure 6.19 shows how kernels of different widths when used for age normalisation affect the classification rate, finally justifying our use of a kernel of 20yrs for our dataset.

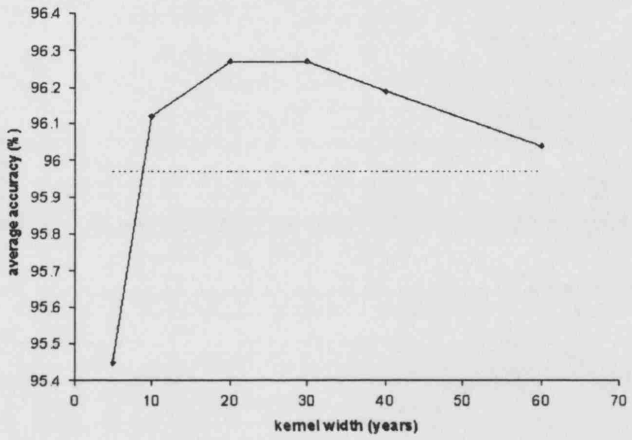


Figure 6.19: The effect on classifying Noonan Syndrome using age normalisation with kernels of different widths. These results are shown for a single fold of the data, to save computation time. The dotted line shows the classification rate that is achieved without age normalisation, as a baseline for comparison. If the kernel is too narrow (5yrs) the trajectory is unreliable, leading to poor classification rates. Likewise if the kernel is too wide the marginal benefit of using age normalisation is lost. The best improvement that is achieved on this fold is about 0.3%, with a width of between 20 and 30 yrs.

6.4 Conclusions

In this chapter we have shown that the extra shape information that is included in a dense surface model is extremely useful for classification tasks. In particular, for screening for Noonan syndrome we have shown that an SVM classifier trained on landmark data alone performs dramatically worse than if the full set of corresponded vertices is used. Similar results have been obtained for other syndromes, including Velo-cardio-facial Syndrome: 88% (Hammond et al., 2004b), Smith-Magenis Syndrome: 93% (Hammond et al., 2003), Williams Syndrome: 94% (Hammond et al., 2004a). These results

demonstrate the immediate clinical application of DSMs for identifying the characteristic face shape associated with certain syndromes and for using those characteristics to screen for the presence or absence of the syndrome in new scans.

Additionally, we have presented a method for analysing continuous variables such as age to obtain averaged trajectories through shape-space. These trajectories can be computed in the absence of longitudinal data by using kernel smoothing to obtain a moving average. Having these trajectories allows us to alter the apparent age of a scan by moving an example along a parallel path to the trajectory, under the assumption that the aging trajectories for individuals are parallel. Having access to longitudinal data would help prove or disprove this assumption, and the data collection for this is ongoing.

Chapter 7

Combined Colour and Surface Models

This chapter looks at how a DSM can be extended to model the grey-level or colour texture of surfaces as well as their shape, giving a photo-realistic face model. We demonstrate that such a model has a potential role in training clinical geneticists to recognise different facial dysmorphic syndromes.

7.1 Introduction

The 3D face scans used in this work have been captured using stereo-photogrammetric scanners, where multiple camera views are processed by stereo matching algorithms into a single 3D surface. The result is a polygonal surface, plus colour or intensity information at every point on that surface. Such a textured surface is typically stored in two parts, the surface and a *texture image* that is projected onto it. This form of data storage is not specific to any brand of scanner but is an efficient way of storing the data, for reasons detailed later in this chapter. Figure 7.1 shows how the textured surface is made.

Textured surfaces are directly supported by hardware rendering on most graphics cards owing to their extensive use in computer games. When sent to the hardware for rendering, each polygon is rendered with a texture by supplying *texture coordinates* (tcoords) for each vertex in addition to its 3D space coordinates:

$$\mathbf{v} = (x, y, z, t_x, t_y) \tag{7.1}$$

Importantly, textured surfaces need not be stored in this form, other higher-level representations are possible, as described below.

By convention, the tcoords run from 0 to 1 horizontally and vertically, with the texture image, whatever its resolution or shape, being mapped into the unit square (Fig. 7.1, centre). For the face data with which we have been working, the same

Figure 7.1: An illustration of how a textured surface (right) consists of a polygonal mesh (left) plus a texture image (centre) that is mapped onto it, triangle by triangle. The texture image can be a composite of multiple views, as here, since there is no requirement for neighbouring triangles in the mesh to be neighbouring in the texture image.

texture image is used for all the polygons in the surface. If this were not the case then the texture images would have to be merged. In games, typically, the same bit of texture image is used to texture many polygons (eg. a brick wall texture) but this is not the case with textured surface scans of the face.

Different ways of texturing are possible. Figure 7.2 shows some common ones. At the top is the most intuitive; a digital photograph of an object is used to texture it. This scan is from an early scanner developed at the 3D-Matic Lab in the Department of Computer Science at the University of Glasgow[1]. Only a limited part of an object can be textured in this way - only those bits of it visible from a particular location.

An improvement over this is afforded by using a cylindrical texture map (Fig. 7.2, middle). This scan is from a Cyberware 3030 head and face scanner which scans around the head on a rotating arm. Here a larger area of the surface can be texture-mapped than with single-view texturing. For human faces this technique works quite well, with only small areas remaining untextured, eg. behind the ears or on top of the head. However, for more complicated objects (a teapot, for example) this approach is problematic.

Arbitrary surfaces can be texture-mapped using multi-view texture images (Fig. 7.2, bottom). This scan is from the Facial Capture System from MedEIM. Unlike the two examples above, such surfaces have the property that neighbouring polygons on the surface may map to polygons that are not neighbours in the texture image. This is depicted in Fig. 7.3 where the surface from (Fig. 7.2, bottom) is rendered in different colours for the two halves of the texture image. In general there may be many views combined into a texture image.

Another illustration of how the polygonal mesh is formed of two halves is shown

[1] `http://www.faraday.gla.ac.uk/`

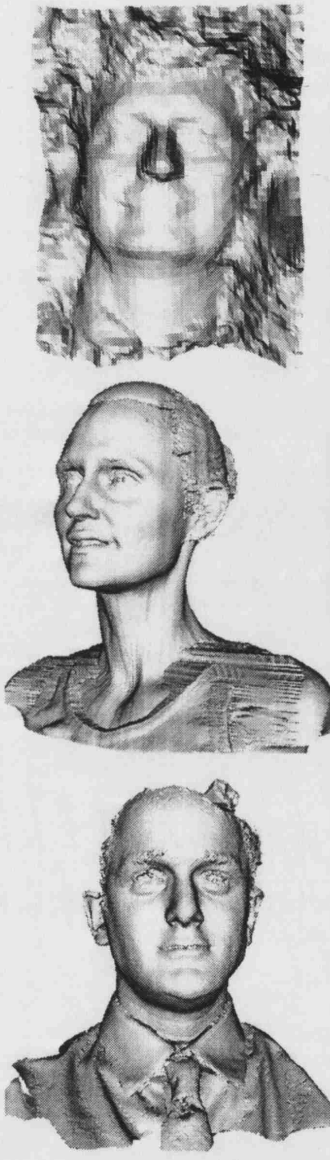


Figure 7.2: Three different texturing methods. Single-view texturing (top) projects a photographic image onto a polygonal surface. Cylindrical texturing (middle) allows more of the surface to be textured. Multi-view texturing (bottom) allows arbitrary surfaces to be textured.

Figure 7.3: A multi-view textured surface (from Fig. 7.2, bottom) with the the different halves of the surface shaded differently. Many of the polygons that are neighbouring in the surface are not neighbouring in the texture image, meaning that vertices that would normally be shared between polygons cannot be since they require different tcoords.

in Fig. 7.4, where the mesh of Fig. 7.3 is projected into the $[0, 1]$ space of the tcoords. This is readily achieved by replacing each vertex's (x, y, z) coordinates with $(t_x, t_y, 0)$. Note that there is a direct correspondence between this image and the texture image in Fig. 7.1 (centre).

The use of multi-view texturing has implications for the way surfaces are stored and rendered. The simplest way for a polygonal mesh to be stored is by *per-vertex texturing*:

A: a list of vertices, each (x, y, z)

B: the faces - a list of tuples of indices into A

C: a list of tcoords, each (t_x, t_y)

List C corresponds with list A, enabling the tcoords for each vertex to be retrieved: vertex $A[i]$ has tcoords $C[i]$. Each vertex is thus effectively stored in the form of (7.1).

For multi-view textured surfaces (Fig. 7.2, bottom), however, this form of representation is not possible. Vertices on the border between the two halves of the surface (see Fig. 7.3) require two sets of tcoords, and in general possibly as many as one for each polygon in which the vertex is used. In the VRML[2] specification the solution is to use *per-face texturing*, where there are *four* lists that define a surface:

[2]Virtual Reality Modelling Language, a text format for specifying 3D objects and simple behaviours. The format specification is at `http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm`

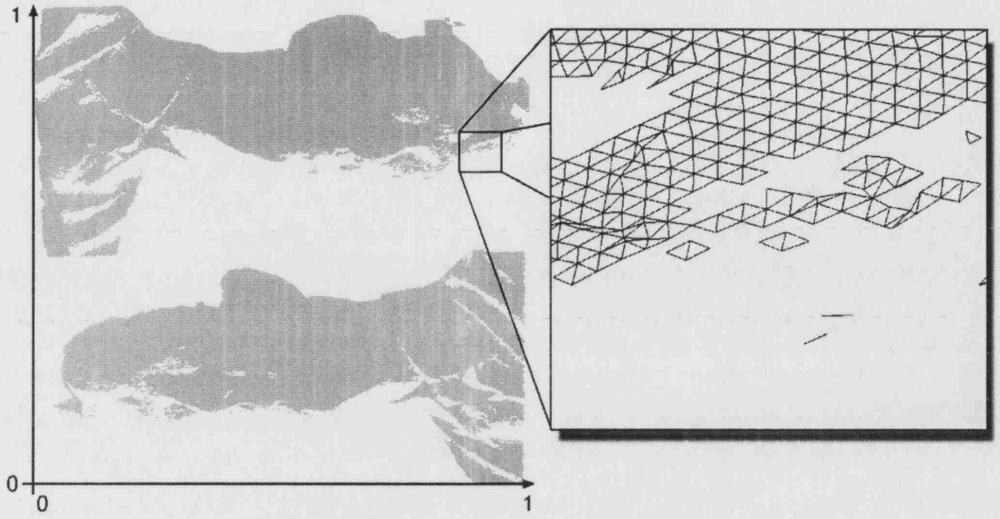


Figure 7.4: The polygonal mesh from Fig. 7.3 projected into the texture-image space. There is a direct correspondence between this space and that shown in Fig. 7.1 (centre).

A: a list of vertices, each (x, y, z)

B: the faces - a list of tuples of indices into A

C: a list of tcoords, each (t_x, t_y)

D: the texture faces - a list of tuples of indices into C

List D corresponds entry-for-entry with list B and thus must be of the same length. Lists A and C, however, need not be of equal length, since some vertices may require more than one set of tcoords. The following scheme is used to extract the vertices and tcoords for each polygon:

- the ith polygon in list B is $B[i]$
- the indices of the vertices are $B[i][0]$, $B[i][1]$, $B[i][2]$, etc. (eg. three of them if the polygon is a triangle)
- the vertices of the polygon are $A[B[i][0]]$, $A[B[i][1]]$, etc.
- the tcoords of each vertex are $C[D[i][0]]$, $C[D[i][1]]$, etc.

In either per-vertex or per-face texturing, the tcoords are interpolated across the polygon to retrieve the colour values for each point on the surface.

Another common format, OBJ[3], also permits per-face texturing, with a similar list structure to VRML. The OBJ format also permits some polygons to remain untextured, an added complication when working with textured surface data as it cannot be assumed

[3] `http://www.dcs.ed.ac.uk/home/mxr/gfx/3d/OBJ.spec`

that every polygon has texture coordinates. For the data we have been using these polygons are few in number and are filtered out before building a textured model.

In general, multi-view textured surfaces need not be stored in a per-face texturing format. One alternative is to convert it to per-vertex texturing by 'flattening' it. This may be necessary before rendering anyway, depending on the graphics library being used. In VTK (which we are using for other reasons) and OpenGL, per-face textured surfaces must be flattened before rendering.

Conversion from a per-face texturing representation to a per-vertex texturing representation is achieved by *vertex duplication*. Figure 7.5 illustrates how this works. This operation is not necessarily irreversible - if the smallest edge length in the surface is ϵ then any vertices closer together than ϵ can be assumed to be duplicates and could be merged at a later date.

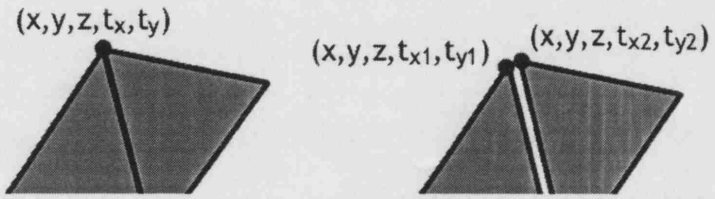


Figure 7.5: Vertices shared between two or more polygons can either be stored as one object (left) or as two (right). Conversion between the format on the left to the one on the right is achieved by duplicating the vertex. While the structure on the left has advantages for storage and surface processing, the one on the right allows different tcoords to be assigned to the vertex, depending on which polygon is being textured. The gap between the triangles on the right is for clarity only, the surface looks identical when rendered since the vertices have not moved.

In the simplest case, every vertex is duplicated. Thus if there are 10,000 triangles, the resulting surface would store 30,000 vertices, regardless of how many of these were duplicates. Such a method of storage has obvious disadvantages: greater storage space is required (and hence longer loading times), and algorithms which rely on vertices being shared (decimation, computation of normals, etc.) cannot be directly applied. However, there is one important advantage: the surface can be manipulated directly in VTK, allowing all the existing functionality for locating points on surfaces, etc., to be utilised. As we shall see, these functions will be essential to building a textured surface model.

The second disadvantage of converting to a per-vertex format, mentioned above, is actually not very significant because algorithms like decimation are only partly effective on per-face textured data anyway. Per-face textured surfaces, such as the one shown in Fig. 7.3, cannot be decimated directly in a simple manner because if polygons across the join are merged the tcoords will be widely spread across the texture image (Fig. 7.1,

centre) and will be wrong. A special decimation algorithm that respected the tcoords would be required to decimate this kind of data correctly, and even then it could not decimate fully since the edges along the join would have to remain untouched[4].

If storage is a problem, a more sophisticated way of converting a surface from per-face texturing to per-vertex texturing is to duplicate only those vertices where the tcoords are different. Polygons neighbouring in Fig. 7.4 could retain their shared vertices since the tcoords of those vertices are the same.

If normals are required on a surface, these are best computed *before* vertex duplication. If the polygons are stored as unconnected units, their normals will be perpendicular to their plane[5], whereas if the surface is connected then vertex normals can be computed as the average normal of each adjacent polygon. Average normals are useful for smooth-rendering (Gouraud, Phong, etc.). However, when rendering with a texture map without lighting correction (all the scans shown) the surfaces are typically rendering with 100% ambient light, to get the best effect. If lighting correction were used (see for example Blanz and Vetter, 1999), and the surface had to be re-lit (Fig. 7.3 shows a simple example of this with the two halves of the surface differently shaded) then normals would be required. Algorithms that use the local orientation of the surface may also require normals, for example to compute the directed distance between two surfaces.

A clear distinction should be drawn between textured surfaces like the ones shown, and volume images (eg. MRI, CT), where intensity or colour information is stored at every point in a 3D grid. Surfaces can be extracted from volume images (typically by contouring) but intensity information is not usually stored on these surfaces (since if the surface had been extracted by contouring it would be equal at all points). If the surface were moved through the volume the intensity could be sampled at each step but this is not possible for a textured surface, where the intensity information is undefined at points not on the surface.

Having clarified the type of data with which we are working, in the following sections we explain how to build textured surface models from a collection of scans.

7.2 Background

Soon after the development of active shape models (Hill et al., 1992; Cootes et al., 1992), the modelling of the shape of structures in images was extended to include their intensity variation: Active Appearance Models (AAMs) were created (Cootes and Taylor, 1994; Cootes et al., 1998). AAMs are an extension of eigenfaces (Turk and

[4]For an approach to decimating textured surfaces correctly, see the work of Hugues Hoppe: `http://research.microsoft.com/~hoppe/`

[5]Non-planar polygons are not normally used in representing surfaces, or for rendering, because they do not represent a unique patch of surface.

Pentland, 1991) - the application of PCA to intensity data (Sirovich and Kirby, 1987) - but take account of the shape and pose variation by producing a 'shape-free' image for each input example (see also Craw and Cameron, 1991).

AAMs can be extended to 3D volume images in a straightforward manner, although requiring large amounts of data storage (Wolstenholme and Taylor, 1999). In general, however, textured surfaces require a different approach.

With texture-mapping methods where the neighbouring-polygon property is maintained, such as single-view and cylindrical texturing, it is reasonably straightforward to find the shape-free texture images when the correspondence between the surfaces has been made. In O'Toole et al. (1995); Vetter et al. (1997); Blanz and Vetter (1999, 2003), the correspondence is established on a cylindrical depth map image (from Cyberware scans) using optical flow, and the same warping is used to bring the texture image into correspondence. In Paterson and Fitzgibbon (2003) the same type of data is warped into correspondence using 30 landmarks and radial basis function (RBF) warping (of which thin-plate splines are a particular example).

There is a limitation to these techniques. As noted earlier, one of the advantages of DSMs is that they can be applied to structures with an arbitrary topology, since there is no requirement that the input surfaces be closed, contiguous or even locally manifold, as long as the topology is reasonably consistent across the dataset. The techniques mentioned above were designed for and only work on a particular type of textured surface data - they cannot incorporate multi-view-textured surface data. If textured scans of complex surfaces were acquired, there would be no way these methods could be used to build a textured surface model from the scans, since the texture would not map to a single contiguous image. The situation can be seen in Fig. 7.7 (top) in the next section, where multi-view textures are shown next to a single-view texture. An RBF warp driven by landmarks cannot map these faces onto each other, since this would require a discontinuity. The main contribution of this chapter is to show how this problem can be solved, essentially by a discontinuous, piece-wise warp of the images.

In this chapter we give details of how textured surface models can be produced from arbitrary surfaces. We also show a clinical application of them.

7.3 Creating shape-free images

To make a PCA model of the intensity and colour variation across a set of surfaces, a pixelwise-correspondence must be established between the texture images in the dataset. Following Cootes et al. (1998), we produce *shape-free* images. When the correspondence of the geometry of the scan with the base mesh is established (as described in Chapter 3), we can use it to resample each texture image into a shape-free version, which has a pixel-wise correspondence with the base mesh texture image. The

steps defining this correspondence and the resampling process are listed and illustrated in Figure 7.6.

This procedure returns a shape-free version of each texture image, which can be mapped correctly onto the base mesh polygons (the ones that are used as the deformable template, see Chapter 3). The stack of these images can then be passed into a PCA as in Cootes et al. (1998). For a simple dataset of three faces, the resampled shape-free images are shown in Fig. 7.7. Eight landmarks (corners of the eyes and mouth, nose tip and chin point) were used to register these three scans.

Some pixels in the image do not map onto any part of the surface, these are left black. Pixels that do not map exactly to the surface, but are close to pixels that do, are given the colour of these neighbouring pixels. This prevents dark edges appearing on the rendered surface owing to the interpolation that may be used when texture-mapping. To achieve this extended mapping, a tolerance of a pixel or two is used in step 1 of the resampling process.

Since the shape-free images are in pixelwise correspondence with the base mesh's texture image, they can correctly be texture-mapped onto the base mesh. Figure 7.8 shows an example of this. When the surfaces are similar in shape this can yield images that look natural, as if a third person had been imaged[6]. However, if the shape differs too much the pattern of lighting will not match the surface shape and the image will look wrong owing to conflicting perceptual clues. Maintaining the correlation between the surface shape and the texture variation is a crucial part of building convincing textured surface models and is explained in the next section.

Curiously, the perception of the middle face in Fig. 7.8 seems to differ from person to person; some see it as more similar to the face on the left, others as more similar to the face on the right. Our perception of face identity is an interesting subject in its own right, software to synthesise faces from computer models is already being marketed for use in psychology and neurological studies (`www.genemation.com`).

7.4 Combining shape and appearance models

Following Cootes et al. (1998), we build separate PCA models of the shape and appearance variation and then combine these models into a single PCA by concatenating the input vectors:

$$\mathbf{w} = \begin{pmatrix} \mathbf{w_s} \\ \mathbf{w_g} \end{pmatrix} \tag{7.2}$$

[6]Our software for doing this has been used by artists in a video piece called "Remote Mind: the strangers are still me" by Georg Mühleck and Barbara Rauch, first shown in Inverness in September 2001.

Steps for resampling a texture image:

1. for each pixel in the base mesh image, find the base mesh polygon onto which it maps (if any) and find the corresponding 3D point on the base mesh TPS-warped to the mean landmarks (computed as part of the DSM algorithm),
2. find the closest point on the example mesh warped to the mean landmarks,
3. find within which polygon this point lies, and to where in the example texture image this point maps,
4. sample the colour at this point from the example texture image and write it to the pixel in the shape-free example image being created at the location from which we started in the base mesh image (leave all non-written pixels black).

Figure 7.6: The resampling process for making shape-free images of every textured surface in a collection.

Figure 7.7: The original texture images (top) for three surface scans and the pixelwise-corresponded shape-free images (bottom) produced by the algorithm illustrated in Fig. 7.6. The first scan is used as the base mesh and determines the layout in the texture image. If a single-view texture map image (as the one shown on the top-right) had been used then all the shape-free images would similarly be a single-view.

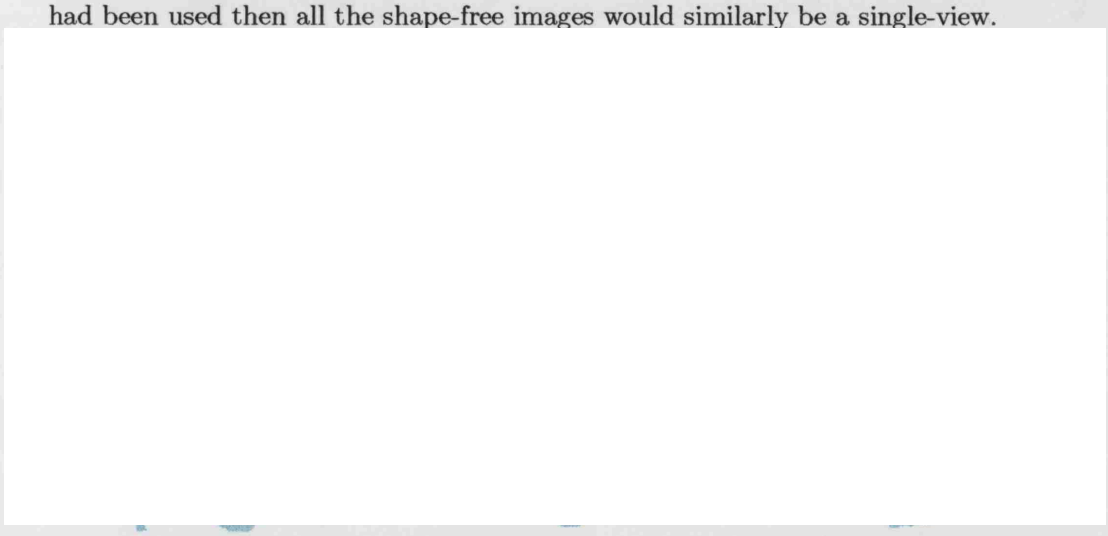


Figure 7.8: On the left and right are two original textured surface scans. In the middle is the surface from the one on the left with the texture from the one on the right mapped onto it. The texture used is the shape-free image shown in Fig. 7.7 (bottom, centre). Note that the features are mapped correctly (eyes onto eyes, etc.), indicating that the correspondence achieved by using TPS-warping and landmarks (8 in this case) is reasonably correct.

where $\mathbf{w_s}$ are the shape mode weights for an example, and $\mathbf{w_g}$ the intensity mode weights.

However, the models are in different units - typically millimeters for the surface and 0-255 for the intensity variation - and so must be made commensurate with a scaling factor. The method given in Cootes et al. (1998) involves altering the shape parameters to move the template landmarks on the image by a small amount and computing the change in the intensity parameters, giving a diagonal matrix of scaling factors. This method does not easily extend to the case of textured 3D surfaces because the texture does not alter if we move the vertices, unlike in 2D images.

In Cootes' online report[7], a simpler method is mentioned - to use the ratio of the total variances to scale one of the models, for example:

$$\mathbf{w} = \begin{pmatrix} k\mathbf{w_s} \\ \mathbf{w_g} \end{pmatrix} \tag{7.3}$$

with

$$k = \sqrt{\frac{\sum \lambda_g}{\sum \lambda_s}} \tag{7.4}$$

where λ_s are the eigenvalues for the shape model and λ_g the eigenvalues for the intensity model. When building models that are later to be combined in this way it makes sense to retain all the modes of variation in the initial shape and intensity models (ie. not reduced to those explaining a fraction, such as 98%, of the shape or intensity variation) since the low-lying modes in one model may be correlated with higher modes in the other model and thus should be retained.

It should be noted that while this method seems to work well in practice and is very easy to compute, there are alternatives. A sensible approach is described in Sumpter et al. (1997), where the eigen-entropy of the resulting model is computed:

$$E = -\sum p_i \log_2(p_i) \tag{7.5}$$

where p_i are the normalized eigenvalues given by

$$p_i = \frac{\lambda_i}{\sum \lambda}. \tag{7.6}$$

The entropy is computed for a range of scale factors, k, and the factor that gives the largest (negative) entropy is chosen since this is where, in some sense, the model retains maximal information. Figure 7.9 shows the entropy against a range of scale factors for a 3D face model, with a peak at $k = 0.12$. In practice the value of k suggested by

[7] `http://www.isbe.man.ac.uk/~bim`

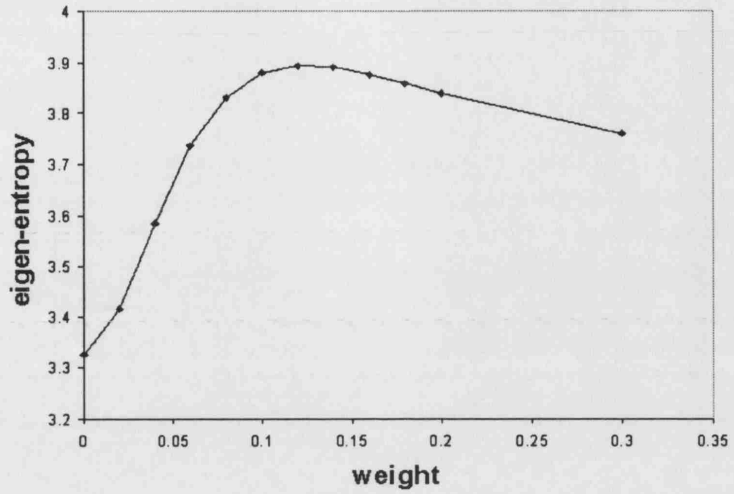


Figure 7.9: A plot of the eigen-entropy of a combined surface shape and texture model of faces, against the scale factor. The maximal entropy is given by a scale factor of 0.12, suggesting that the model built with this factor will contain the most information.

this more expensive method is similar, though not identical, to that computed by the simple method of (7.4).

The motive behind this approach is related to that used in the MDL approach (Davies et al., 2002a,b), namely that measures of the information content of the model can be used to determine the optimal parameters for building it.

7.5 Results

In Fig. 7.10 we show the first three principal components of a combined model computed using the simpler method of (7.4). This model was computed using scans of 51 subjects with Williams Syndrome[8], with ages between 4 months and 44 years.

Such a textured model has a clinical application in the training of clinical geneticists to identify visually those dysmorphic syndromes that have an associated facial abnormality. These syndromes can be very rare, making it difficult for individual clinicians to learn the features characteristic of the syndrome. Williams Syndrome (WS), for example, occurs in only once in every 20,000 births. Without having seen many children with the syndrome it can be difficult for a clinical geneticist to separate the features that are particular to the syndrome from the natural variation in face shape. The synthesised faces in Fig. 7.10 are averages of a whole collection of faces with WS and thus average-out those facial features that vary across a population, making these faces a very attractive method for training inexperienced clinical geneticists to spot

[8]For more information on Williams Syndrome, see `http://www.williams-syndrome.org`

WS.

Another way to visualise the facial features associated with a dysmorphic syndrome is to build a textured model including both positives and controls, and then to morph between the average of each class by animating a sequence of frames from one to the other. In a dynamic display it is easy to spot the features that are changing. Such a morph is easily obtained by visualising faces from along the line in shape-space that joins the averages. The features can even be exaggerated by moving further along the same line, although of course the face will start to look very odd if regions outside the normal range of the distribution are sampled. Such animations can be found in the online supplementary material for Hammond et al. (2004b), at `http://www.mrw.interscience.wiley.com/suppmat/0148-7299/suppmat/2004/126/v126.339.html`

Figure 7.11 shows faces from along an aging trajectory computed in a textured surface model, using the method presented in section 6.3 (p. 75). The changes associated with aging are much clearer in this sequence than in the non-textured surfaces shown in section 6.3 - in later life the face can still be seen to change, unlike that in Fig. 6.11 (p. 82).

7.6 Discussion

Synthesising textured surfaces given a location in the combined shape-and-intensity-space is an efficient process. The shape-free image with the appropriate weights is simply a weighted sum of the eigenvectors and the mean. Similarly, the surface shape can be directly retrieved. To render the textured surface, the graphics engine takes care of the mapping necessary, often with hardware support, and so the image does not need to be warped in any way. In practice, surfaces can be rendered at several frames per second without any difficulty.

Figure 7.8 suggests a possible method for guiding the correspondence of the dense surface model, namely to use the colour information to ensure that the features of the face are mapping onto each other. This approach was used in Blanz and Vetter (1999) with optical flow to correspond cylindrically-textured face scans. One problem with this is that optical flow, since it is obtained from local computations, can often be erroneous and optical flow fields frequently require post-processing, whilst our method is global and might produce a more reliable correspondence. For example, if the eyebrows have been altered by plucking, the correspondence obtained by optical flow would be different to that obtained by our method, and for diagnosing medical conditions such as dysmorphic syndromes this kind of variation is not usually desirable. In general, a good test of the quality of a correspondence is to evaluate the performance of classifying facial types in unseen scans.

Figure 7.10: Synthesised faces from a textured surface model of 51 faces with Williams Syndrome. The rows are the top three principal components, in order of decreasing variance. Each row shows faces between -3 standard deviations (left) and +3 standard deviations (right).

Figure 7.11: Frames from along the age trajectory from a textured model.

We *could* run the classification tests (Chapter 6) using the textured models themselves, and in fact it seems likely that the performance would improve. However, without a controlled light-source we would not be able to state confidently that there was no correlation between the lighting conditions under which each scan was taken and the presence or absence of the syndrome we were trying to classify. A classification system that utilised texture or appearance could thus be classifying on the difference in the lighting, rather than on the features of the syndrome. Typically, face scans are acquired in batches, often in different rooms. If a set of scans with Williams Syndrome, for example, were all taken on one day in a single session, the lighting condition in the room where the scans were taken would be confounded in a textured surface model with the shape changes associated with WS. By testing with cross-validation we would not necessarily be able to show that the system would continue to work well in a clinical setting where the lighting is unknown.

By contrast, we are reasonably confident that the surface shape is invariant to different lighting conditions, since acquistion systems usally control the lighting to a sufficient degree to be able to reliably acquire a surface. There is an issue that scans from different scanners have different resolutions and drop-out patterns but the resampling steps of the DSM algorithm should avoid these getting into the final model. Additionally, many of the artefacts peculiar to particular scanners are not correlated with the facial features. For example, striping is often visible in laser scans (see for example Fig. 7.2, centre left) but unless the stripes always appear at a certain point on the face this shape variation will be relegated to the insignificant shape modes.

The fitting of the deformable model to an unseen scan (Chapter 5) could in theory be improved by making use of the texture information as well as the surface shape information. In particular, if the search procedure were expressed as a generic energy minimisation problem then error terms for the texture mismatch (after sampling the texture from the target surface) could be added without difficulty. However, it is not clear what the improvement would be; conceivably, the texture information would help to disambiguate lips from chins for example.

Another use of the textured model would be to fit the 3D deformable template to a 2D image. This application is explored in Blanz and Vetter (1999), Paterson and Fitzgibbon (2003), and elsewhere. Ideally some form of lighting correction is required in order to be able to re-light the face to match the lighting in the scene being fitted. Such a face-tracking system has the potential to be more robust than 2D or $2\frac{1}{2}$D models (eg. Cootes et al., 2000) since it can represent the rotation and lighting variation of a face in a native fashion. In particular, effects such as specular highlights and sharp shadows on the face can be modelled directly. The textured surface model presented here could be used for this purpose but this has not been explored.

Depending on the scanner used, some of our scans are in RGB colour while some are in greyscale only. Our current implementation of texture models reduces colour images to a single grey-level channel as a preprocessing step but it is a trivial modification to build a colour texture model. Obviously if a colour model is to be built then every scan in the dataset must be colour or the variation between grey-looking faces and colourful ones would appear in the model as a major mode of variation.

The technique of combining multiple PCA models into one is useful in general, and could be used, for example, to merge shape models of different parts of the body. By ensuring that the variation in each model is commensurate (either by the eigen-entropy method, or the simpler ratio-of-variances method) we can merge models from different modalities, for example 2D with 3D, intensity with shape, greyscale with colour, or volume with surface. Clinically it might be beneficial to merge, for example, face scans of subjects with landmarked x-rays of their hand, since the growth of the hand is sometimes used as a measure of growth stage in children, and limb abnormalities are not uncommon in genetically-based conditions.

7.7 Conclusions

The contribution of this chapter is that we have shown a method for building textured surface models of arbitrarily-shaped surfaces, including those which aren't topologically equivalent to a flat sheet. While for human faces it may be possible to achieve the same results in other ways (eg. by using a cylindrical mapping), the ability to cope with multi-view textured surfaces means that the same technique can be used as the extent of the captured area increases.

Chapter 8

Conclusions

8.1 Summary of contributions

We have shown (Chapter 3) that it is possible to build densely-corresponded surface models of the human face from raw surface scans that include holes, noise, and a great variation in the amount of surface covered. To do this, a small set of hand-placed landmarks is required. The surfaces are brought into correspondence by thin-plate spline (TPS) warping the faces to a sparse set of hand-placed landmarks on a base mesh, and this mesh is used to resample the surfaces. The surfaces are trimmed automatically by imposing a threshold on the remaining separation between the surfaces after the TPS-warping step.

In Chapter 5 we detailed a method by which DSMs can be used to automatically register unseen face scans. A hybrid of iterated closest point (ICP) and active shape model (ASM) fitting was used to move a template and to deform it within limits consistent with a training population to best match the target. An evaluation of fitting to 21 scans showed that the landmarks were automatically placed within an average (across 10 landmarks) RMS error of 3.0mm. Experiments are being undertaken to determine how this compares with human landmark placement error on similar scans. Initial results suggest that the errors are of the same magnitude Gwilliam (2004).

Chapter 6 demonstrated that using dense surface models (DSMs) gives far better performance at classifying face shape than landmarks alone. This is because the surface in between and beyond the landmarks contains a lot of shape information that is relevant to the classification problems we have tested. This finding motivates the clinical use of DSMs for screening for facial dysmorphic syndromes, an application which is now being actively pursued.

Also in this chapter we presented a method for computing average growth trajectories through shape-space, even where longitudinal data is not available. Having a good model of the non-linear effects of age on the face is crucial for understanding both normal growth patterns and how those growth patterns can sometimes go wrong to

make dysmorphic features.

Finally, in Chapter 7 we showed that it is possible to build textured surface models of arbitrarily-shaped surfaces, by using the dense correspondence to make shape-free images of each textured scan. Textured surfaces improve the visualisation of the features associated with the syndromes we have studied, and might potentially be used to improve the classification, although good lighting correction would probably be required for this.

8.2 Future work

There are many alternatives and extensions to the method presented, some of these are discussed here.

8.2.1 Making the dense correspondence

One possible extension would be to explore different methods of making the correspondence between the surfaces. Finding the closest point to each vertex in the base mesh is arguably the simplest method. Alternatives include mesh regularisation (Lorenz and Krahnstöver, 2000) and Markov random field smoothing (Paulsen et al., 2002; Hilger et al., 2004). The classification problem is a suitable test-bed for comparing different methods.

Perhaps a more interesting possibility would be to find some method for applying the information-theoretic approach of Davies et al. (2002a,b) (see also Thodberg, 2003a,b) to surface scans that include holes and a great variety in extent captured (Fig. 1.2, p. 9). Currently the approach has been demonstrated on open and closed curves in 2D and on closed surfaces in 3D. The method could be extended to open surfaces in 3D by using constraints to prevent the area being modelled from collapsing to a single point. Either some method for filling the holes reliably or a way of parameterising the surfaces even in the presence of holes would be required. Additionally, some method for specifying what area was to be included would be necessary, perhaps by using a hand-crafted base mesh as a master example.

8.2.2 Bootstrapping the model

One possibility for reducing the work required to landmark all the scans is to *bootstrap* the model by using it to fit to the new examples being added. In Chapter 5 we presented a method that could achieve comparable accuracy in landmark placement to a human, this could be used to landmark unseen scans before recomputing the model using them. Much experimentation would be required to see how robust this procedure was; whether the errors that it would introduce would accumulate excessively. (Note that this is a

different use of the word 'bootstrapping' than that used on page 80.)

Bootstrapping could also potentially be used to improve the correspondence of the existing examples in the training set. A computed model could be used to fit to the members of the training set, and the model could then be recomputed. Again thorough experimentation would be needed to check the stability of this iterative process.

8.2.3 More on growth trajectories

The modelling of growth patterns in Chapter 6 has particular clinical significance for certain genetic conditions and there is plenty of scope for more work in this area. In some facial dysmorphic syndromes it has been suggested that the dysmorphic features are due to the *timings* of the growth processes being different to that in normal growth, in addition to the shape changes introduced by under- or over-development in different areas (Allanson et al., 1985). If specific mutations in the genes known to have an impact on facial growth could be correlated with the timing of different growth processes then a clearer picture of the genetic causes of facial dysmorphology would emerge.

As a first step, enough data needs to be acquired on individual syndromes in order to compute their growth trajectories. A comparison of such a trajectory with that for the controls would give an indication not only of how the face is typically different at different stages of growth but at what point the trajectory diverges most rapidly. For some conditions where surgical operations are used to improve the facial appearance, having such a picture of the timings of the growth of the face would assist in deciding when to operate.

With enough data, the kernel width used to compute the average growth trajectories (Chapter 6) could be reduced, yielding better trajectories. It is hoped that age normalisation using these improved trajectories would improve the performance of the classification for syndromes.

As mentioned in Chapter 6, if longitudinal data were available then the trajectories for individuals could be visualised in comparison to the average trajectory. A set of such individual trajectories would give an idea how predictable growth was from the shape of the face - a smooth vector field of trajectories would allow us to infer the future path of the face shape through shape-space based on the position. It seems likely that even with a little longitudinal data we could improve the performance of algorithms for estimating age and for predicting future appearance.

8.2.4 Family resemblence

Having a good model of how a face ages would open up another avenue of possibilities: predicting the appearance of children's faces from those of their parents. With enough scans of parents and biological children, all of known ages, it should be possible to

work out how predictable the appearance of a child is from the faces of its parents. One possibility would be to use the age model to normalize the age of each parent and child in the training set, and then the distribution of the examples could be visualised and analysed to identify any patterns. For example, the children might form a Gaussian distribution centred on the mean of their parents. Having scans of the grandparents, or other information, might improve the prediction.

The clinical relevance of such work would be to separate family resemblence as one of the factors affecting face shape. Identical twins have been used to help identify genetic syndromes because in the rare instances where one twin has a mutation and the other not then the effect of the mutation on the face can be directly identified. However, such instances of twins are exceedingly rare, especially in genetic syndromes that are themselves rare. If a model of the familial appearance could be computed from scans of the family of an individual with a certain syndrome, then perhaps these effects could be usefully removed from the individual's face, and hence the features associated with the syndrome could be identified. It seems likely that this would work better if the immediate family was large, to mitigate the effects of individual variation.

8.2.5 Decimating the surface

Throughout this thesis we have tended to avoid the issue of mesh decimation (Schroeder et al., 1992; Hoppe, 1996), where the resolution of a polygonal mesh is reduced to save on storage and time requirements without changing the shape of the surface too much. This was for two reasons. Firstly, it is difficult to decimate multi-view textured surfaces (see Ch. 7) since not every edge can be collapsed without producing errors in the texturing. However, it is possible with some decimation algorithms to specify hard constraints on the process; to forbid an edge from being collapsed, or to take into account the texture image (Garland and Heckbert, 1998; Sander et al., 2001).

More interestingly, textured surfaces aside, the issue of in what order to collapse the edges becomes more difficult to answer when we are considering shape models. Decimation proceeds by preferring to collapse those edges that lie in flat areas of the surface, since more vertices are needed in more highly curved parts of the surface to represent the shape well. But in our model the shape of the surface is not a fixed property, since different shape modes deform the surface in different ways. It is conceivable that there could be some method for taking the shape model into account when decimating, perhaps by choosing to collapse those edges in regions that remained mostly flat throughout the entire volume of shape-space. Whether the savings of decimation would be worth the effort of computing such a solution remains to be shown.

8.2.6 Facial expression

The shape of a face is determined by many factors, including gender, age, identity, ethnic background, syndrome, etc. One important factor we have not yet considered is facial expression; whether the person is smiling or frowning, whether they are speaking at the time the scan was taken, etc. In two dimension there have been studies of the variation in face appearance of an individual (Bettinger et al., 2002; Costen et al., 2002) that have modelled the regions in shape-space occupied exclusively by an individual. It would be interesting to apply these methods to our registered 3D face scan data.

One interesting possibility for understanding how the face moves is to capture dynamic sequences using stereo photogrammetry hardware and compute the 3D surface from each frame. Experiments with data supplied by `www.surfim.com` that were captured in this way are ongoing. As with the aging, dynamic sequences give a trajectory in shape-space along which the face moves. If the complete range of movement could be captured this might result in a useful tool for morphing photo-realistic textured faces for animation purposes. An analysis of the correlation between words spoken and face shape would allow photo-realistic data-driven 'talking heads' to be created (Kuratate et al., 1998, see also `http://www.haskins.yale.edu/haskins/HEADS/contents.html`).

8.2.7 Models of separate parts of the face

One interesting alternative to building shape models of the entire surface of the face is to look at specific regions such as the eyes, the nose and the mouth. By comparing the classification performance of these 'sub-models' for Noonan-control screening, for example, we could see which parts of the face are most discriminating. One application of this that is currently being pursued is to look at individuals who have an atypical deletion for a particular syndrome. By classifying such faces using sub-models we hope to be able to identify who has particular features of a syndrome (for example, the eyes) but not others.

To do this, we manually edit the base mesh to restrict it to the region of interest, using a standard mesh editing tool. The same set of landmarks can be used for these submodels since they just control the TPS warp used to bring the surfaces into correspondence. Figure 8.1 shows some of the submodels we have tried.

One idea not yet explored is to look at these submodels with respect to age. If we were to compute the average trajectories for syndromic faces versus controls using these submodels we should be able to identify which parts of the face were changing most at different stages.

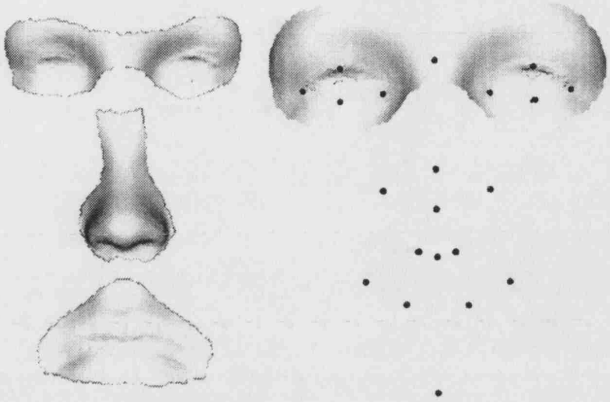


Figure 8.1: An illustration (left) of the different components of the face for which surface models can be built individually. On the right is the average surface computed for the eyes submodel, along with the landmarks used.

8.2.8 Multi-class classification

In Chapter 6 we considered two-class classification but if there are more than two classes to which an example might belong then the issue of classification becomes more complicated. For the application of screening for syndromes it is essential that the system not be restricted to just identifying whether a scan has a given syndrome or not. Ideally the system would return an ordered list of the most likely syndromes, with an associated probability for each.

Some classifiers are capable of inferring a multi-class division of the input space directly (Li et al., 2003). SVMs can be used for multi-class classification by pairwise or all-against-one classification (Hsu and Lin, 2002; Wu et al., 2003). The more recent work (Wu et al., 2003) also suggests a method for pairwise classification that is shown to be more stable than using voting.

An illustration of how the input space can be carved into multiple decision regions is given in Fig. 8.2. Multi-class classification has not yet been tested on our face data but would be very useful for screening purposes. Loos et al. (2003), for example, successfully applied multi-class classification to a set of photos of children with different syndromes.

8.2.9 Fitting the textured 3D model to 2D images

One application of textured surface models (Chapter 7) is to automatically locate human faces in 2D images. This application in explored in Blanz and Vetter (1999, 2003); Paterson and Fitzgibbon (2003). A three-dimensional representation of the face is a more natural representation than 2D or 2.5D models (Cootes and Taylor, 2000) since sharp shadows and specular highlights can be directly synthesised to match the target. Face images where the pose is extremely unusual might also benefit from having a full

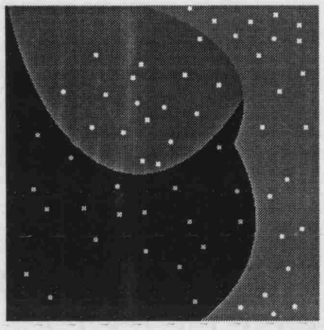


Figure 8.2: An illustration of how the decision regions for a three-class problem in two dimensions might look.

3D model of the target object.

Lighting correction is necessary if the target face is to be modelled correctly, since the lighting in the target image can vary enormously and must be synthesised along with the pose, facial expression and identity to get a good match. A good method for correcting the lighting of our database of surface scans might make it possible to use texture to improve the classification ability.

8.2.10 Security and surveillance applications

An application that we haven't explored is the classification of face shape by individual identity. Three-dimensional models of the face potentially offer a more robust facial recognition solution since for example, unlike 2D systems, they cannot be fooled by an intruder holding up a photograph. Making a 3D model of someone is much harder, assuming their 3D scan is kept secure.

Face biometrics are already being included in the ID cards being introduced around the world. Companies that sell 3D face recognition solutions include Geometrix (`www.geometrix.com`).

8.3 Final conclusions

We have presented *Dense Surface Models*, a technique for building surface models from surface scans. The technique is immediately useful because it can take as input the type of data typically produced by laser-scanning or stereo-photogrammetry surface scanners, including holes and spikes and a great variety in the extent of the area captured. We have built models using almost 2000 such scans of volunteers. By modelling the entire surface, and not just key landmark points, the model is very good at discriminating between different groups, such as children that have a certain syndrome and those that do not. This has led to our software being trialled for screening purposes as part of an internationally funded project.

Bibliography

Allanson, J., Hall, J., Hughes, H., Preus, M., and Witt, D. (1985). Noonan syndrome: the changing phenotype. *Am. J. Med. Genet.*, 21:507–514.

Andresen, P., Bookstein, F., Conradsen, K., Ersbøll, B., Marsh, J., and Kreiborg, S. (2000). Surface-bounded growth modeling applied to human mandibles. *IEEE Trans. Med. Imag.*, 19(11):1053–1063.

Audette, M., Ferrie, F., and Peters, T. (2000). An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 4(3):201–217.

Besl, P. and McKay, N. (1992). A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.*, 14:239–256.

Bettinger, F., Cootes, T., and Taylor, C. (2002). Modelling facial behaviours. In *Proc. British Machine Visionm Conference*, pages 797–806.

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *SIGGRAPH'99 Conference Proceedings*, pages 187–194.

Blanz, V. and Vetter, T. (2003). Face recognition based on fitting a 3D morphable model. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(9):1063–1074.

Bookstein, F. (1997a). Landmark methods for forms without landmarks: Localizing group differences in outline shape. *Medical Image Analysis*, 1(3):225–243.

Bookstein, F. (1997b). Shape and the information in medical images: A decade of the morphometric synthesis. *Computer Vision and Image Understanding*, 66(2):97–118.

Brett, A., Hill, A., and Taylor, C. (1997). A method of 3D surface correspondence for automated landmark generation. In *British Machine Vision Conference*, pages 709–718. BMVA.

Brett, A. and Taylor, C. (1998). A method of automated landmark generation for automated 3D PDM construction. In *British Machine Vision Conference*, pages 914–923. BMVA.

Brett, A. and Taylor, C. (2000). Construction of 3D shape models of femoral articular cartilage using harmonic maps. In Delp, S., DiGioia, A., and Jaramaz, B., editors, *MICCAI2000*, pages 1205–1214. Springer-Verlag.

Burbidge, R. (2002). Adaptive kernels for support vector classification. In *Proc. MSRI Workshop on Non-linear Estimation and Classification*, pages 345–356. Springer-Verlag.

Carr, J., Beatson, R., Cherrie, J., Mitchell, T., Fright, W., McCallum, B., and Evans, T. (2001). Reconstruction and representation of 3D objects with radial basis functions. In *Proc. SIGGRAPH'01*, pages 67–76.

Chang, C.-C. and Lin, C.-J. (2001). LIBSVM: A library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cootes, T., Edwards, G., and Taylor, C. (1998). Active appearance models. In Burkhardt, H. and Neumann, B., editors, *Proc. European Conference on Computer Vision*, volume 2, pages 484–498. Springer.

Cootes, T., Hill, A., Taylor, C., and Haslam, J. (1994). The use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–366.

Cootes, T. and Taylor, C. (1994). Modelling object appearance using the grey-level surface. In *Proc. BMVC94*, pages 479–488. BMVA Press.

Cootes, T. and Taylor, C. (1997). A mixture model for representing shape variation. In Clark, A., editor, *British Machine Vision Conference*, pages 110–119.

Cootes, T. and Taylor, C. (2000). Combining elastic and statistical models of appearance variation. In *Proc. European Conference on Computer Vision, Vol. 1*, pages 149–163.

Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1992). Training models of shape from sets of examples. In Hogg, D. and Boyle, R., editors, *Proc. British Machine Vision Conference*, pages 9–18. Springer-Verlag.

Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1995). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Cootes, T., Wheeler, G., Walker, K., and Taylor, C. (2000). Coupled-view active appearance models. In *Proc. British Machine Vision Conference*, pages 52–61.

Costen, N., Cootes, T., Edwards, G., and Taylor, C. (2002). Automatic extraction of the face identity-subspace. *Image and Vision Computing*, 20:319–329.

Craw, I. and Cameron, P. (1991). Parameterising images for recognition and reconstruction. In *British Machine Vision Conference*, pages 367–370, London. Springer-Verlag.

Davies, R., Twining, C., Cootes, T., Waterton, J., and Taylor, C. (2002a). 3D statistical shape models using direct optimisation of description length. In Heyden, A., Sparr, G., Nielsen, M., and Johansen, P., editors, *Proc. 7th European Conference on Computer Vision*, volume 3, pages 3–20, Copenhagen, Denmark.

Davies, R., Twining, C., Cootes, T., Waterton, J., and Taylor, C. (2002b). A minimum description length approach to statistical shape modeling. *IEEE Trans. Med. Imag.*, 21(5):525–537.

Davison, A. and Hinkley, D. (1997). *Bootstrap Methods and Their Application.* Cambridge University Press, Cambridge.

Dean, D., Hans, M., Bookstein, F., and Subramanyan, K. (2000). Three-dimensional Bolton-Brush growth study landmark data: Ontogeny and sexual dimorphism of the Bolton standards cohort. *The Cleft Palate - Craniofacial Journal*, 37(2):145–156.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., and Vapnik, V. (1997). Support vector regression machines. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9, page 155. The MIT Press.

Dryden, I. and Mardia, K. (1998). *Statistical Shape Analysis.* Wiley.

Ericsson, A. and Åström, K. (2003). Minimizing the description length using steepest descent. In *Proc. British Machine Vision Conference, Norwich, United Kingdom*, volume 2, pages 93–102.

Fawcett, T. (2003). ROC graphs: Notes and practical considerations for data mining researchers. Technical report, HP Laboratories, Palo Alto, CA, USA. Tech report HPL-2003-4. `http://www.hpl.hp.com/personal/Tom_Fawcett/papers/`.

Feldmar, J. and Ayache, N. (1994). Rigid, affine and locally affine registration of free-form surfaces. Technical report, INRIA. no. 2220.

Fitzgibbon, A. (2001). Robust registration of 2D and 3D point sets. In *Proc. British Machine Vision Conference*, pages 411–420.

Garland, M. and Heckbert, P. S. (1998). Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings of the Conference on Visualization '98*, pages 263–269. IEEE Computer Society Press.

Gerig, G., Styner, M., Jones, D., Weinberger, D., and Lieberman, J. (2001). Shape analysis of brain ventricles using SPHARM. In *Proc. Mathematical Methods in Biomedical Image Analysis*, pages 171–178, Kauai, Hawaii.

Ghattaura, A. (2001). Analysing facial form and motion using geometric morphometrics. Master's thesis, University College London.

Goldgof, D., Lee, H., and Huang, T. (1988). Motion analysis of nonrigid surfaces. In *IEEE Proc. Conference on Computer Vision and Pattern Recognition*, pages 375–380.

Goodall, C. (1991). Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B*, 53(2):285–339.

Gower, J. (1975). Generalized procrustes analysis. *Psychometrika*, 40:33–51.

Guéziec, A. and Ayache, N. (1994). Smoothing and matching of 3D space curves. *Intl. J. Computer Vision*, 12(1):79–104.

Gwilliam, J. (2004). Reproducibility of 3d landmark placement. Master's thesis, Eastman Dental Institute, University College London. (to appear).

Hammond, P., Hutton, T., Allanson, J., Buxton, B., Campbell, L., Karmiloff-Smith, A., Murphy, K., Patton, M., Pober, B. Smith, A., and Tassabehji, M. (2004a). 3D dense surface models identify the most discriminating facial features in dysmorphic syndromes. In *Proc. American Society for Human Genetics.* (submitted).

Hammond, P., Hutton, T., Allanson, J., Campbell, L., Hennekam, R., Holden, S., Patton, M., Shaw, A., Temple, I., Trotter, M., Murphy, K., and Winter, R. (2004b). 3D analysis of facial morphology. *American Journal of Medical Genetics, Part A*, 126(4):339–348.

Hammond, P., Hutton, T., Allanson, J., and Smith, A. (2003). The 3D face of smith-magenis syndrome (SMS): a study using dense surface models. *Eur. J. Hum. Gen.*, 11(S1):102. Proc. European Human Genetics Conference.

Hammond, P., Hutton, T., Patton, M., and Allanson, J. (2001a). Delineation and visualisation of congenital abnormality using 3D facial images. In Bellazzi, R., Zupan, B., and Liu, X., editors, *Proceedings of the Workshop Intelligent Data Analysis in Medicine and Pharmacology, IDAMAP2001 at MedInfo2001*, pages 26–29, London, UK.

Hammond, P., Hutton, T., Patton, M., and Allanson, J. (2001b). Use of 3D photogrammetry in the craniofacial assessment of Noonan syndrome. In *XXII David W. Smith Workshop on Malformations and Morphogenesis*, pages 97–100, UCLA Conference Centre, Lake Arrowhead, CA, USA.

Hilger, K., Paulsen, R., and Larsen, R. (2004). Markov random field restoration of point correspondences for active shape modelling. In *Proc. SPIE Medical Imaging.* (to appear).

Hill, A., Cootes, T., and Taylor, C. (1992). A generic system for image interpretation. In Hogg, D. and Boyle, R., editors, *Proc. British Machine Vision Conference*, pages 276–285. Springer-Verlag.

Hoppe, H. (1996). Progressive meshes. In *Proc. SIGGRAPH'96*, pages 99–108.

Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Networks*, 13:415–425.

Huber, P. (1981). *Robust statistics.* Wiley, New York.

Hutton, T., Buxton, B., and Hammond, P. (2001). Dense surface point distribution models of the human face. In *Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 153–160, Kauai, Hawaii.

Hutton, T., Buxton, B., and Hammond, P. (2002). Estimating average growth trajectories in shape-space using kernel smoothing. In Sporring, J., Niessen, W., and Weickert, J., editors, *Proc. International Workshop on Growth and Motion in 3D Medical Images*, pages 1–7, Copenhagen, Denmark.

Hutton, T., Buxton, B., and Hammond, P. (2003a). Automated registration of 3D faces using dense surface models. In Harvey, R. and Bangham, J., editors, *Proc. British Machine Vision Conference*, pages 439–448, Norwich.

Hutton, T., Buxton, B., Hammond, P., and Potts, H. (2003b). Estimating average growth trajectories in shape-space using kernel smoothing. *IEEE Trans. Med. Imag.*, 22(6):747–753.

Johnson, R. (1963). On the theorem stated by Eckart and Young. *Psychometrika*, 28:259–263.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. International Joint Conference on Artificial Intelligence*, pages 1137–1145.

Kuratate, T., Yehia, H., and Vatikiotis-Bateson, E. (1998). Kinematics-based synthesis of realistic talking faces. In *Internatinal Conference on Auditory-Visual Speech Processing (AVSP'98)*, pages 185–190.

Lanitis, A., Taylor, C., and Cootes, T. (2002). Towards automatic simulation of aging effects on face images. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(4):442–455.

Lee, A. W. F., Dobkin, D., Sweldens, W., and Schröder, P. (1999). Multiresolution mesh morphing. *Computer Graphics Proceedings (SIGGRAPH 99)*, pages 343–350.

Lester, H., Arridge, S., Jansosn, K., Lemieux, L., Hajnal, J., and Oatridge, A. (1999). Non-linear registration with the variable viscosity fluid algorithm. In *Image Processing and Medical Imaging (IPMI99)*, Visegrad, Hungary.

Li, T., Zhu, S., and Ogihara, M. (2003). Using discriminant analysis for multi-class classification. In *Third IEEE International Conference on Data Mining*, pages 589–592.

Lindstrom, P. and Turk, G. (1998). Fast and memory efficient polygonal simplification. In *Proc. Visualization '98*, pages 279–286. IEEE Computer Society Press.

Loos, H., Wieczorek, D., Würtz, R., von der Malsburg, C., and Horsthemke, B. (2003). Computer-based recognition of dysmorphic faces. *European Journal of Human Genetics*, 11:555–660.

Lorensen, W. and Cline, H. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21:163–169.

Lorenz, C. and Krahnstöver, N. (2000). Generation of point-based 3d statistical shape models for anatomical objects. *Computer Vision and Image Understanding*, 77:175–191.

MacLeod, N. (2001). Landmarks, localization, and the use of morphometrics in phylogenetic analysis. In Edgecombe, G., Adrain, J., and Lieberman, B., editors, *Fossils, phylogeny, and form: an analytical approach*, pages 197–233. Kluwer Academic/Plenum, New York.

Maintz, J. and Viergever, M. (1998). A survey of medical image registration. *Medical Image Analysis*, 2:1–36.

Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal. Soc. Indust. Applied Math.*, 11(2):431–441.

Moghaddam, B. and Yang, M.-H. (2002). Learning gender with support faces. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(5):707–711.

Morris, R., Kent, J., Mardia, K., Aykroyd, R., Fidrich, M., and Linney, A. (1999a). Exploratory analysis of facial growth. In *Proc. CISST*, Las Vegas.

Morris, R., Kent, J., Mardia, K., Fidrich, M., Aykroyd, R., and Linney, A. (1999b). Analysing growth in faces. In *Proc. Leeds Annual Statistical Workshop*, Leeds.

O'Higgins, P. and Jones, N. (1998). Facial growth in cercocebus torquatus: an application of three-dimensional geometric morphometric techniques to the study of morphological variation. *Journal of Anatomy*, 193(2):251–272.

O'Higgins, P., Jones, N., Ghattaura, A., Hammond, P., Hutton, T., and Carr, M. (2002). Geometric morphometric approaches to the study of soft tissue growth and expression in the human face. *American Journal of Physical Anthropology*, 117:119. Suppl. S34.

O'Toole, A., Vetter, T., Bülthoff, H., and Troje, N. (1995). The role of shape and texture information in sex classification. Technical Report 23, Max-Plank-Institut für biologische Kybernetik. http://www.mpik-tueb.mpg.de.

Paterson, J. and Fitzgibbon, A. (2003). 3D head tracking using non-linear optimization. In *British Machine Vision Conference*, pages 609–618.

Paulsen, R. and Hilger, K. (2003). Shape modelling using markov random field restoration of point correspondences. In Taylor, C. J. and Noble, J. A., editors, *Proc. Information Processing in Medical Imaging*, volume 2732 of *Lecture Notes in Computer Science*, pages 1–12. Springer.

Paulsen, R., Larsen, R., Laugesen, S., Nielsen, C., and Ersbøll, B. (2002). Building and testing a statistical shape model of the human ear canal. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002, 5th Int. Conference, Tokyo, Japan.* Springer.

Pentland, A. and Horowitz, B. (1991). Recovery of non-rigid motion and structure. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(7):730–742.

Rogers, M. and Graham, J. (2002). Robust active shape model search. In Heyden, A., editor, *Proc. European Conference on Computer Vision*, pages 517–530.

Rueckert, D., Frangi, A., and Schnabel, J. (2003). Automatic construction of 3D statistical deformation models of the brain using nonrigid registration. *IEEE Trans. Med. Imag.*, 22(8):1014–1025.

Sander, P., Snyder, J., Gortler, S., and Hoppe, H. (2001). Texture mapping progressive meshes. In *Proc. SIGGRAPH'01*, pages 409–416.

Schölkopf, B., Smola, A., and Muller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Schölkopf, B., Smola, A., Williamson, R., and Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, 12:1207–1245.

Schroeder, W., Martin, K., and Lorenson, W. (1997). *The Visualization Toolkit (2nd ed.)*. Prentice-Hall, New Jersey. `http://www.vtk.org`.

Schroeder, W., Zarge, J., and Lorensen, W. (1992). Decimation of triangle meshes. In *Proc. SIGGRAPH'92*, pages 65–70.

Sirovich, L. and Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *J. Optical Soc. Am*, 4:519–524.

Sozou, P., Cootes, T., Taylor, C., DiMauro, E., and Lanitis, A. (1997). Non-linear point distribution modelling using a multi-layer perceptron. *Image and Vision Computing*, 15:457–463.

Subsol, G., Thirion, J., and Ayache, N. (1994). Non rigid registration for building 3d anatomical atlases. In *Proc. IEEE International Conference on Pattern Recognition*, pages 576–578.

Sumpter, N., Boyle, R., and Tillett, R. (1997). Modelling collective animal behaviour using extended point distribution models. In *Proceedings of the British Machine Vision Conference*, pages 242–251. BMVA.

Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293.

Tagare, H. (1999). Shape-based nonrigid correspondence with application to heart motion analysis. *IEEE Trans. Med. Imag.*, 18(7):434–439.

Tartaglia, M., Mehler, E., Goldberg, R., Zampino, G., Brunner, H., Kremer, H., van der Burgt, I., Crosby, A., Ion, A., Jeffery, S., Kalidas, K., Patton, M., Kucherlapati, R., and Gelb, B. (2001). Mutations in PTPN11, encoding the protein tyrosine phosphatase SHP-2, cause Noonan syndrome. *Nature Genetics*, 29:465–468.

Thirion, J. (1994). Extremal points: definition and application for 3D image registration. In *IEEE Proc. Conference on Computer Vision and Pattern Recognition*, pages 587–592.

Thodberg, H. (2003a). Adding curvature to MDL shape models. In Harvey, R. and Bangham, J., editors, *Proc. British Machine Vision Conference*, pages 251–260, Norwich.

Thodberg, H. (2003b). Minimum description length shape and appearance models. In Taylor, C. J. and Noble, J. A., editors, *Proc. Information Processing in Medical Imaging*, volume 2732 of *Lecture Notes in Computer Science*, pages 51–62. Springer.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Üzümcü, M., Frangi, A., Reiber, J., and Lelieveldt, B. (2003). The use of independent component analysis in statistical shape models. In Sonka, M. and Fitzpatrick, J., editors, *Proc. SPIE Medical Imaging*, pages 375–383.

Vapnik, V. (1995). *The nature of statistical learning theory.* Springer, New York.

Vetter, T., Jones, M., and Poggio, M. (1997). A bootstrapping algorithm for learning linear models of object classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 40–46.

Wang, Y., Peterson, B., and Staib, L. (2000). Shape-based 3D surface correspondence using geodesics and local geometry. In *Computer Vision and Pattern Recognition*, volume 2, pages 644–651, Hilton Head Island, South Carolina.

Wolstenholme, C. and Taylor, C. (1999). Wavelet compression of active appearance models. In *Proc. MICCAI*, pages 544–554.

Wu, T.-F., Lin, C.-J., and Weng, R. (2003). Probability estimates for multi-class classification by pairwise coupling. In *Proc. NIPS 2003.* `http://www.csie.ntu.edu.tw/~cjlin/papers/svmprob/svmprob.pdf`.

Zhang, Z. (1992). On local matching of free-form curves. In *Proc. British Machine Vision Conference*, pages 347–356.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comp. Vision*, 13(2):119–152.